

AN INVESTIGATION OF CERTAIN
PRODUCTION PLANNING PROBLEMS IN A
FLEXIBLE MANUFACTURING ENVIRONMENT

By

MELTEM DENIZEL SIVRI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1993

Copyright 1993

by

Meltem Denizel Sivri

To the loving memories of my parents, Meliha Denizel and Muzaffer Denizel.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to Dr. S. Selcuk Erenguc, chairman of my supervisory committee. Without his continual guidance and encouragement this work could not have been completed.

I would also like to thank Dr. Harold Benson for his invaluable comments and Dr. Gary Koehler for his continual support. I wish to thank Dr. Chung Yee Lee, Dr. Patrick Thompson, and Dr. Li-Hui Tsai for making themselves available in my supervisory committee.

I would like to thank my friend Serpil Sayin for sharing with me the worst and the best of this experience. Finally I would like to thank my husband Mustafa Sivri for his understanding and encouragement.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
ABSTRACT	vii
CHAPTERS	
1 INTRODUCTION	1
2 LITERATURE SURVEY	8
2.1 Introduction	8
2.2 FMS Planning Problems	14
2.2.1 Identification of Subproblems	14
2.2.2 Hierarchical Approaches	16
2.2.3 Part-Type Selection Problem	19
2.2.4 Machine Loading Problem	27
2.2.5 Other Planning Problems	40
2.3 FMS Scheduling	42
2.4 Summary	52
3 FMS PLANNING MODELS	57
3.1 Model 1: A Part-type Selection and Lot-sizing Model	58
3.1.1 Single Machine Case	58
3.1.2 A Generalization of Model 1 : Multi-machine Case	62
3.2 Model 2: A Part-type Selection and Lot-sizing Model For Minimizing Inventory and Backorder Costs	64
3.2.1 Single Machine Case	64
3.2.2 A Generalization of Model 2 : Multi-machine Case	65
3.3 An Inclusive Model for Part-type Selection, Lot-sizing and Machine Loading	67
3.4 A Heuristic Procedure	70
4 SOLUTION PROCEDURES	74
4.1 Model 1	74

4.1.1 A Lower Bounding Scheme	74
4.1.2 Algorithm ALG1	86
4.1.2.1 An informal description of algorithm ALG1	87
4.1.2.2 A formal statement of algorithm ALG1	89
4.2 Model 2	91
4.2.1 A Lower Bounding Scheme	91
4.2.2 Algorithms ALG2 and ALG3	93
4.2.2.1 An informal description of algorithm ALG2	93
4.2.2.2 A formal statement of algorithm ALG2	95
4.2.2.3 An informal description of algorithm ALG3	96
4.2.2.4 A formal statement of algorithm ALG3	97
4.3 Model 3	98
5 COMPUTATIONAL RESULTS	103
5.1 Model 1	103
5.1.1 Computational Results for Algorithm ALG1	103
5.1.2 Heuristic Procedure	112
5.2 Model 2	115
5.2.1 Computational Results for Algorithm ALG2	117
5.2.2 Algorithm ALG3 : Implementation Issues	125
5.2.3 Comparison of ALG2 and ALG3	126
5.2.4 Heuristic Procedure	128
6 CONCLUSION AND FURTHER RESEARCH	132
REFERENCES	137
BIOGRAPHICAL SKETCH	148

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

AN INVESTIGATION OF CERTAIN
PRODUCTION PLANNING PROBLEMS IN A
FLEXIBLE MANUFACTURING ENVIRONMENT

By

Meltem Denizel Sivri

May 1993

Chairman: S. Selcuk Erenguc

Major Department: Decision and Information Sciences

In this dissertation we investigate certain production planning problems in a flexible manufacturing system. We present five mixed integer programming models to consider part-type selection, lot sizing and machine loading problems. They incorporate two different demand situations over a T-period planning horizon. For each demand situation we start with a single machine formulation for part-type selection and lot sizing problems. Machine time and tool magazine capacities are aggregated to represent overall system restrictions. Subsequently, we generalize both formulations to consider the machines in the system separately. The generalized formulations also incorporate machine loading problem in terms of part-type assignment. We then present an inclusive

model which considers machine loading in terms of operation assignments. For the single machine formulations we present branch and bound algorithms which provide optimal or near optimal solutions to the problems. These algorithms generalize also to solve the multi-machine problems. For the inclusive model we give a heuristic procedure. Since it is designed for the most general model, this procedure is also applicable to the other models. We report some computational results for the branch and bound algorithms and the heuristic procedure. This dissertation also includes a review of literature on flexible manufacturing systems with special emphasis on the planning problems.

CHAPTER 1 INTRODUCTION

Flexible Manufacturing Systems (FMS) consists of numerically controlled (NC) machines linked by automated material handling devices integrated under the control of a central computer. They are designed to efficiently handle mid to low volume production for a diverse range of products. FMSs can broadly be classified into Flexible Flow Systems (FFS) and General Flexible Machining Systems (GFMS) [Rachamadugu and Stecke, 1989]. These systems can be viewed as the counterparts of traditional flow shops where the parts flow is usually unidirectional and job shops where there is a multidirectional flow of parts. The nature of demand is an important factor in the identification of these systems. FMSs can be identified as dedicated or nondedicated systems according to the variety and volume of part-types they process. A dedicated FMS is usually a part of a larger manufacturing environment. Hence a specific set of part-types is processed in relatively large volumes in order to meet a downstream demand. In nondedicated FMSs a random variety of part-types is processed according to customer orders. In either case the main objective is to complete the processing of the demanded part-types by their due dates which may be imposed by an assembly schedule of a downstream assembly line or by customer orders.

In an FMS environment production planning involves the selection of part-types to be manufactured in a planning period and the allocation of various resources among

the selected part-types within that period so as to meet their due dates. When a part-type is selected its operations are assigned to one or more machines and the required tools are loaded on the appropriate tool magazines. Since the part-types have to be mounted on fixtures before they are released to the FMS, their fixture assignments also need to be made. Assignment of operations to machines will determine the routing of a part-type through the system and it is possible to have fixed or alternative routings for the part-types. These attributes make the FMS production planning problems more complicated than their counterparts in conventional manufacturing systems.

In the literature various approaches have been developed to tackle the FMS planning problems. A common approach is to identify the relevant subproblems and solve them separately in an iterative framework. Stecke [1983] identifies five FMS planning subproblems as part-type selection, machine grouping (forming groups of identically tooled machines), production ratio determination, allocation of pallets and fixtures, and machine loading. A major drawback of this approach is the lack of sufficient integration between the subproblems which may lead to infeasibilities when the overall problem is considered. An alternative approach is to deal with the FMS planning problem in its entirety. The main disadvantage of this approach is its excessive computational requirements. However, we believe that an integrated consideration of the problem as a whole in a model which captures the relationships between various subproblems will provide a better understanding of its structure. Furthermore, such an overall approach eliminates the need for the explicit consideration of some of the subproblems since these will be imbedded in the integrated problem. As stated in Rajagopalan [1986], forming

groups of identically tooled machines, decreases the flexibility for performing diversified operations. Therefore, instead of considering a machine grouping subproblem, it is less restrictive to let the machine groups be formed while operations and their required tools are being assigned to machines. Similarly selection of part-types can be made together with machine loading, so that part-types are selected only if their operations can be performed on the FMS, thus eliminating the possibility of an infeasible selection. We also believe that the FMS loading problem should be considered dynamically throughout a planning horizon. This will provide a better utilization of the system resources by balancing the workload among periods in the planning horizon.

For an FMS with real world dimensions, an integrated formulation of the planning problem in a dynamic manner will result in a large scale mathematical programming problem which will, in all likelihood, have excessive computational requirements for obtaining an optimal solution. However, considering the problem in its entirety is of both theoretical and practical value. Our objectives are to develop computationally efficient heuristic procedures based on the integrated formulation which yields "good quality" solutions; develop exact solution procedures which can be used to obtain exact solutions for relatively small problems; and to verify the quality of the heuristic solutions vis-a-vis an exact solution. In developing exact procedures, "good" lower bounding techniques need to be devised. By a good lower bounding technique we mean, either one that gives a tight bound with some computational effort or one that yields a not so tight bound but requires relatively little computational effort. These lower bounding techniques can also be used for verifying the quality of the heuristic solutions without

resorting to exact procedures. Solutions obtained from a heuristic procedure based on the integrated model would appear to be of more practical value than the solutions obtained by solving the subproblems independently. In addition, the inherent structure of such a model may allow a natural subproblem decomposition that can be used as the basis of an efficient solution procedure. In this manner the links between the subproblems can be preserved and exploited for the purpose of obtaining a "good" solution.

The Mixed Integer Programming (MIP) models we present incorporate the following attributes of the FMS planning problem. We consider the problem over a T-period planning horizon. Depending upon the nature of the environment where it is implemented, the length of the planning horizon may be a week or a month and accordingly the periods are days or weeks. In the literature, the most commonly cited planning criterion is the maximization of production rate. In order to achieve this, operational objectives such as workload balancing are defined. However, according to our observations and the findings of a survey presented in Smith et al. [1986], most FMS users consider meeting due dates to be, if not the most, one of the most important criteria. Therefore we decided to use this criterion as a basis of our planning decisions. We consider two different demand characteristics which give rise to two different models. First we assume that, in the beginning of the planning horizon there is a set of part-types each having a demand specified for the whole planning horizon and to be satisfied as soon as possible. Some part-types may have priorities over others due to their due dates or their relative importance. The objective function we propose to use

facilitates the selection of more urgent part-types in earlier periods during the planning horizon. In the second case, we assume that the demands for the part-types, in each period of the planning horizon, are specified. The objective is to produce the demanded amount of a part-type just-in-time. To achieve this we highly penalize any backorders. Carrying inventories of part-types is allowed. This serves the purposes of increasing system utilization in periods of low demand and avoiding possible backorders in future periods of high demand. However, there is a cost associated with holding inventory.

For these two different demand situations, we study single-machine models which deal with the part-type selection and lot sizing problems and multi-machine models which also incorporate the machine loading problem in terms of part-type assignments. In the single machine models there are two types of resources; available machine capacity of each type and available tool magazine capacity on each machine type. We do not bring any restrictions on the available number of cutting tools of each type; however, it can easily be incorporated into the model. In the multi-machine models we also consider fixture allocations by treating fixtures as special machine types. For the second demand situation, we also study the case where machine loading is considered in terms of operation assignments. In a given period, a part-type is selected only if all its operations are assigned to at least one machine together with the required tools. The lot sizes of the part-types are determined according to available capacities of the machines to which their operations are assigned. Each operation of a part-type is processed at an equal amount which is less than or equal to the total demand of that part-type during the planning horizon. Operations of a part-type can be assigned to different machines and the lot size of an operation can be split among machines.

In all these models, we assume that tool magazines of all machines are set up at the end of a period according to the next period's processing requirements. In an FMS, the tool magazine set-up is a main issue even if there is an automatic tool interchange device that can handle tool changes automatically. Some NC machines require fine tuning during tool changes and the time needed for tuning operations is significant relative to the processing times [Tang and Denardo, 1988a]. Taking out all cutting tools in all magazines and replacing these with a new set of tools can take almost a shift [Stecke and Kim, 1988]. Therefore, assuming one tool magazine set-up per planning period is appropriate when the processings of the selected part-types require a high degree of machine utilization within that period. In the FMS presented in Stecke [1988], the tool magazines are set-up at the end of each day according to next day's processing requirements and to meet these requirements the FMS requires 80% utilization on the average. In one of our visits to an FMS facility, we observed tool magazine set-ups that were as infrequent as several weeks. In such environments, the assumption of one tool set-up per period is very realistic. Still, if in a period total processing time requirements of part-types are considerably lower than available machine times on all machines, then additional tool magazine set-ups may be required within that period. In this case, the resulting production schedule can be shifted appropriately before its implementation.

Part-type selection and machine loading problems have been studied independently in the literature. Furthermore, these problems have been solved for fixed order quantities disregarding the lot sizing aspect of the planning problem. Our approach considers part-type selection and machine loading problems in an integrated manner and

also incorporates the lot sizing decisions. As discussed in Hwang and Shogan [1989], order quantity and due date are major attributes of a part-type. Our models take these attributes into explicit account. Another distinguishing feature of our approach is its dynamic consideration of the FMS planning problem over a planning horizon. Naturally, the single period model is a special case of the models considered here.

In the next chapter we give a survey of the FMS literature with particular emphasis to the studies on the part-type selection and the machine loading issues. In Chapter 3, we present MIP formulations for the FMS planning problems under the two demand scenarios. These formulations consider the part-type selection and the lot sizing problems simultaneously. We then discuss generalizations of these formulations which incorporate the machine loading problem. Chapter 3 also includes a heuristic procedure for the most general model. Chapter 4 is devoted to solution procedures devised to efficiently solve these problems. These procedures consist of branch and bound algorithms based on Linear Programming (LP) and Lagrangian relaxations. We report the computational results for both the exact and the heuristic procedures in Chapter 5. Conclusions and a discussion of future research are presented in Chapter 6.

CHAPTER 2 LITERATURE SURVEY

2.1. Introduction

FMSs have evolved in response to a growing need for finding effective solutions to problems of batch manufacturing. In the mid to late 1960s, the changes in the nature of consumer demand towards more specialized and diversified products pushed manufacturers to work with mid to low volumes in job-shop type environments due to their inherent flexibilities. However, the high productivity of transfer lines could no longer be achieved. The introduction of NC machines followed by Computer Numerical Control (CNC) machines and Direct Numerical Control (DNC) systems and later the rapid improvements in the computer technology made it possible to design and implement integrated systems. Hence it became feasible to achieve mid to low volume production to satisfy the diversified and specialized consumer demand with the flexibility of a job-shop and the productivity of a transfer line. The history of development and the current state of FMSs, including their basic components and properties, are presented and discussed in Talavage and Hannam [1988], and an overview and bibliography is given in Huang and Chen [1986]. Today, FMSs are being designed and implemented in many industrialized countries of the world where batch production constitutes about 75% of the metal-worked products. A survey of the present status of FMSs in various countries is presented in Dupont-Gateland [1985]. She discusses different types of flexibilities

involved in FMSs and classify them into three categories as flexible modules and units, flexible transfer lines, and unaligned flexible systems. Smith, Ramesh, Dudek, and Blair [1986] present results of a survey of FMSs conducted in the U.S. and discuss their characteristics. An important result of their study is that most companies in the U.S. consider due dates to be the most important criterion in their planning and scheduling activities. A survey on FMSs implemented in Japan along with their present status and trends is given in Ohmi, Ito, and Yoshida [1984], and Margirier [1987] presents results of a survey of flexible automated machining in France.

With the availability of sophisticated technology, the managers are now faced with more complicated decision problems. The design, justification and effective implementation of FMSs require good management based on sound and precise decisions. There has been a considerable amount of research since late 1970s on various problems inherent in FMSs, starting with strategic design related issues down to operational scheduling decisions. Several classification schemes for FMS problems have been suggested. Stecke [1983] identifies four classes of FMS problems:

1. design problems such as determining the range of part-types to be produced, specifying types of flexibilities required, deciding on the types and capacities of machine tools and material handling devices, and specifying the FMS layout,
2. planning problems such as selection of part-types for immediate production, machine grouping, allocation of pallets and fixtures, machine loading,
3. scheduling problems for determining the sequence with which the part-types will be processed on each machine,
4. control problems related to monitoring the system for handling disturbances, determining preventive maintenance schedules, designing inspection policies.

Kiran and Tansel [1986] give a similar classification. They view FMS problems in five phases:

1. design,
2. aggregate planning,
3. system set-up,
4. scheduling,
5. control.

Here the aggregate planning phase establishes the links between the FMS and the factory-wide production plans. The system set-up problems are the planning problems in Stecke's classification. Kusiak [1985b] identifies two classes of problems:

1. design,
2. operational.

The operational problems are then analyzed in a four level hierarchy which involves forming part and machine groups, machine loading and scheduling. The classification of Suri and Whitney [1984] is from an organizational point of view and consists of three levels:

1. Long term (months/years): design problems,
2. Medium term (days/weeks): planning or set-up problems,
3. Short term (minutes/hours): scheduling and control problems.

Based on an organizational perspective, they give a structural description of a Decision Support System (DDS) required for effective and efficient FMS utilization.

The FMS problems stated above have been approached in the literature with OR methods and lately with artificial intelligence (AI) and expert system techniques. The long term FMS design problems involving decisions such as finding an optimal system configuration, determining the range of part-types which the system will be able to manufacture have been approached mostly by evaluative performance models like

simulation, queuing theory, perturbation analysis, and petri nets. Suri [1985] gives an overview of evaluative models for FMSs. Hybrid models which combine optimization models with queuing network or simulation approaches have also been used to solve design related problems. Brown, Chan, and Rathmill [1985] give an integrated FMS design procedure involving three stages; planning, design and implementation. In the rest of this introductory section we give a brief review of the work on FMS design. Since our research interest lies mainly in the medium-term planning problems our survey of design problems will be limited in scope. Surveys of FMS literature on design problems are given in Kouvalis [1989], Kalkunte, Sarin, and Wilhelm [1986].

Suri [1985] reports the earliest queuing model of an FMS to be CAN-Q developed by Solberg [1977]. It is in the form of a closed queuing network (CQN) and requires some simplifying assumptions, such as no machine blocking, and exponential service times which are more likely to be deterministic. As a consequence, the performance results it provides, such as work station utilizations, queue length distributions, average waiting times, average transit and process times, are highly aggregated. However, it was successfully implemented to model existing FMSs. Later Suri and Hildebrant [1984] developed another approach based on mean value analysis of queues which can handle additional FMS features such as multiple part classes. Buzacott and Shantikumar [1980] use open and closed network models to analyze various FMS designs such as systems having local storage with both finite and infinite capacity and systems having common storage. They use production capacity as the performance measure. Yao and Buzacott [1985a] model the performance of FMSs with limited storage at each work station as an

open queuing network with general service times. Their main assumption is the nonblocking of work stations which is achieved by releasing all finished jobs immediately to be recirculated to a central storage. Throughput times, mean waiting times and the mean number of jobs are used as performance measures. A CQN model to evaluate an FMS in which part routings depend on the system's state is developed in Yao and Buzacott [1985b]. They again make the nonblocking assumption and use the time reversibility of the resulting Markov Process to obtain product form solutions. Yao and Buzacott [1987] derive product form solutions for a class of FMSs that have reversible parts routing. In Yao and Buzacott [1986a] an exponentialization approach for FMSs with unlimited buffer capacity and general service time distributions is presented. This is done by transforming the corresponding CQN into an exponential one. They compare their results for throughput and mean waiting times against simulation results and show that the exponentialization approach performs accurately. Subsequently, Yao and Buzacott [1986b] studied FMSs with limited buffers using a set of CQN models. Based on the operation characteristics of the system, either the fixed-routing or the fixed loading or the dynamic routing model is applied. These models assume exponential service times and lead to accurate results when the local buffer sizes are substantially smaller than total part population.

CQN models of FMSs have been applied in several existing systems. Seidmann, Shalev-Oren, and Schweitzer [1986] give an analytical review of computerized CQN models with emphasis on their relative modeling capabilities and their relevance to FMS design.

As mentioned earlier, hybrid models have also been used to model FMS design problems. One such model which combines mathematical programming with a CQN model of an FMS is presented in Vinod and Solberg [1985]. The objective is to find an optimal FMS configuration. The problem is formulated as the minimization of a cost function subject to a nonlinear throughput constraint. The cost function is defined in terms of unit operating costs and capital investments. The throughput function is evaluated through the use of CAN-Q, the CQN model of Solberg [1977]. Heungsoon, Srinivasan, and Yano [1989] refer to a similar problem in which they also consider the optimal work allocation. Dallery and Frein [1986], Shantikumar and Yao [1987] study similar problems.

Design problems related to the layout of FMSs are generally formulated as quadratic assignment problems with the objective of minimizing material handling costs. There are also graph theoretical approaches to the problem. Kouvelis [1989] discusses various optimal seeking and heuristic procedures for FMS layout design.

In this dissertation we are particularly interested in the medium term FMS problems which are identified as the planning problems in Stecké [1983a], as the system set-up problems in Kiran and Tansel [1986], and as a part of the operational problems in Kusiak [1985b]. We are going to use the terms "planning" and "system set-up" interchangeably to refer to these problems. In the sequel, we give a review of the FMS literature on planning problems and also of the FMS scheduling literature as it relates to the FMS planning issues.

2.2 FMS Planning Problems

2.2.1. Identification of Subproblems

The use of more sophisticated technology makes the FMS planning and scheduling decisions more complex than those of conventional manufacturing systems. The integration of versatile machine tools with automated material handling devices under the control of a central computer provides various types of flexibilities. Stecke, Dubois, Browne, Sethi, and Rathmill [1983] identify eight types of flexibilities which may be applicable in different FMSs. The ability of machine tools to perform a large variety of operations provides product and process flexibilities. The capacity of the machines to hold many tools and their ability to switch from one tool to another very quickly reduce the set-up times between operations dramatically. The automated handling devices perform the load/unload operations quickly and precisely. Reduction of the set-up times due to tool and/or part switches provides volume flexibility making it possible to work with small batch sizes. The complete computer control over all the system activities provides on line information about the system status and therefore make it easier to handle various disturbances. Also due to the routing flexibility of parts and the flexibility in the ordering of their operations, scheduling decisions should become easier. However, this flexibility, if not managed properly, may result in very poor system performance. Jaikumar [1986 p. 69] indicates that "with few exceptions, the flexible manufacturing systems in the United States show an astonishing lack of flexibility. In many cases, they perform worse than the conventional technology they replace. The technology itself is not to blame; it is management that makes the difference."

The planning decisions pertain to issues of system set-up to ensure the operation of the system for a predetermined amount of time without any intervention. These require the solution of complex planning and scheduling problems. To obtain a systemwide solution, it is most desirable to solve these problems in an integrated manner. Unfortunately, this is not computationally feasible. A common approach is to define the basic subproblems and try to solve them separately and sequentially in an iterative manner. Stecke [1983b] defines five FMS planning subproblems:

1. part-type selection,
2. machine grouping,
3. production ratio determination,
4. allocation of resources (pallets and fixtures),
5. machine loading.

Among these subproblems part-type selection and machine loading have attracted considerable research interest. Some authors combined part-type selection with determining production ratios and loading. Yet others considered resource allocation together with loading. Machine grouping has been dealt with both separately and combined with loading. A detailed review of these studies is given in the following sections.

Other authors make alternative classifications of the FMS set-up problem. Kiran and Tansel [1986] identify four subproblems:

1. part-type selection,
2. tooling,
3. fixture allocation,
4. operations assignment.

They formulate an integer programming (IP) model to analyze the connections between these. Their model assigns the operations of the selected part-types to machines while

also making the fixture allocations. The objective is to maximize the number of selected part-types. The model is extended to consider the number of part movements among machines and the number of fixture changes. Kiran [1986] shows that such a formulation of the FMS set-up problem is NP-complete.

2.2.2. Hierarchical Approaches

Because of its inherent complexity, hierarchical approaches have been suggested to tackle the FMS planning problem. These treat the subproblems separately but in an integrated manner. Stecke [1986] presents one such approach for solving the machine grouping and loading problems. First, at an aggregate level a CQN model is used to partition the machines into groups with the objective of maximizing expected production. Then at a detailed level nonlinear mixed integer programming (MIP) models are proposed to solve the loading problem. Another hierarchical framework is presented in Kusiak [1986a] for FMS planning and scheduling. There are four levels in this hierarchy: aggregate planning, resource grouping, disaggregate planning and scheduling. In both of these papers, details of the models and/or solution procedures are not discussed but rather referred to the authors' earlier work which we will discuss later in the following sections. Chakravarty and Shtub [1986] handle the problem in three levels. The first level plan determines the production and inventory amounts together with the shortages for each part-type in each period. In addition, the processing time of each tool on each machine is determined. At the second level, part-tool groups are formed based on the amount of tool sharing between part-types and considering the number of pallets and fixtures and tool magazine capacities. Then, based on the groups formed and assuming

sequential loading of groups onto the FMS, a make-span minimization problem over all groups is solved. For the operational plan a real time scheduling procedure assigns the part-types with the highest percentage of remaining processing time to the machine which becomes available.

Jaikumar and Wassenhove [1987] too present a hierarchical planning framework consisting of three levels. First the quantities of part-types to be processed in a planning period together with the inventories held are determined. The second level assigns the selected part-types to families considering the degree of tool sharing so as to maximize machine utilizations. The third level is concerned with detailed scheduling and control where the use of cyclical schedules is suggested.

A ten step iterative algorithm is developed by Chung and Lee [1986] for selecting part-types and determining their production batches. Starting with a monthly plan first weekly and then daily production schedules are determined. Part-types are selected according to their priorities based on aggregate machine time and tool magazine capacities. Then, an MIP model is solved to determine the part routes that minimize make-span. The resulting routes are checked for feasibility in terms of tool magazine capacities. Subsequently, a goal programming (GP) model is solved to minimize the deviations from the due dates and production requirements.

Kim and Yano [1989b] present an iterative algorithm for solving part-type selection, machine grouping, and machine loading problems simultaneously. They develop a strategy to integrate procedures that are designed to solve these problems

independently. They state that their approach provides "very good" solutions in a "reasonable" amount of time.

The hierarchical model proposed by Mazzola, Neebe, and Dunn [1989] integrates FMS production planning into a closed loop material requirements planning (MRP) environment. The three levels of the hierarchy are FMS/MRP rough cut capacity planning, machine grouping and loading, and detailed scheduling. In rough cut capacity planning, the feasibility of the master production schedule with respect to the FMS capacity is determined. Mazzola [1989] formulates this problem as a generalized assignment problem.

Integration of FMS planning and scheduling problems within the framework of an expert system is presented in Solot [1990]. In this two level hierarchical approach, first current system status is evaluated, then a planning module is employed to determine the part-types to be introduced in the system. Solot suggests use of the flexible approach of Stecke and Kim [1986] (see Section 2.2.3) to tackle this problem. Next step is the establishment of a scheduling period and the development of an appropriate schedule. The "inference rule" is used to help select the appropriate scheduling algorithms according to the current system status. This concept of combining OR methods with the techniques of artificial intelligence and expert systems is illustrated in reference to an existing FMS in Switzerland. The same FMS is also addressed in Bastos [1988]. Bastos identifies four planning functions:

1. batching,
2. routing,
3. dispatching,
4. sequencing.

These are based on a list of part-types provided by a tactical plan which usually covers a week. The batching and routing problems are formulated as linear programming (LP) models which are solved in a sequential and iterative framework.

2.2.3. Part-type Selection Problem

Problem definition: Part-type selection within a short-term planning framework is concerned with the problem of deciding which subset of part-types in the production orders will be processed simultaneously during the coming period. Since each part-type requires a set of tools, this decision involves determining which tools should be loaded onto the machines' tool magazines. The main limitation is the tool magazines which are capable of holding only a limited number of tools. If the FMS is equipped with automatic tool handling devices which can switch tools while the machines are working, then part-type selection will not be an issue. However, since most of the existing FMSs do not have this technology, the part-type selection problem is still a very relevant problem.

Being one of the system set-up problems, part-type selection can either be formulated and solved separately or it can be combined with some other planning problems. In the literature, examples of both approaches exist. When treated separately, it is usually the first problem to be solved in the sequence. Subsequent planning problems are based on the results of part-type selection. A very important aspect inherent in the choice of part-types is meeting the associated due dates. However, since due date considerations make the problem, which is already NP-complete, more

complicated, most of the research on part-type selection does not include any due date related criteria. Nevertheless, we believe that meeting the due date is a major concern of today's FMS users and therefore any realistic problem formulation involving part-type selection must consider some measure of due date. There are basically two approaches to the problem; the batching approach and the flexible approach.

Batching approaches: In the batching approach to the part-type selection problem, once a subset of parts is selected, the required tools are loaded onto the tool magazines and the production process continues until all parts complete their requirements. Then the system is set-up for the next batch. The formulations in Hwang [1986], Hwang and Shogan [1989], Hirabayashi et al. [1984], Rajagopalan [1986], Whitney and Gaul [1985], Tang and Denardo [1988b] all follow this approach. In all these models, there are constraints which account for the tool magazine capacities and constraints that guarantee the loading of appropriate tools when a part-type is selected.

Hwang [1986] formulates the problem as a binary program (BP) with the objective of maximizing the number of parts in a batch. This is a greedy objective which selects the part-types that require the least number of tool slots on the magazine. There is always a chance that some part-types will never be selected. Hwang reports that problems of size about 50 parts and 100 tools, can be solved in a reasonable amount of time with a branch and bound algorithm using Lagrangian relaxation. The same model is considered in Hwang and Shogan [1989] with a revised objective function to consider the due dates and an additional constraint on the total available machine time. Here the objective function is a weighted sum of part-types where the weight of a part-type is a

function of its due date. Lagrangian relaxation is again the suggested solution technique and two relaxations are examined. In one of these, tool magazine capacity and total machine time availability constraints are relaxed leading to a maximal network flow problem. In the other, the constraints defining the part-tool relations are relaxed and the resulting problem consists of two independent 0-1 knapsack problems. The experimental results show that the knapsack relaxation gives better solutions than the network relaxation and it also consumes less computer time.

Hirabayashi, Suzuki, and Tsuchiya [1984] give a very similar formulation in which the part-tool relations are represented by a bipartite graph. The problem is called the optimal tool module design problem. The objective function, as in Hwang and Shogan's [1989] formulation, is the maximization of a weighted sum of the number of selected part-types. Here the weights are the profits associated with the part-types. A branch and bound procedure, which employs a primal-dual algorithm for solving the subproblems, is developed.

In the approaches discussed above, the problems have to be solved sequentially until the production orders set is covered thoroughly. This may be advantageous since it gives an opportunity to include the new orders in the selection. However without any due date considerations this may lead to solutions where certain part-types are never selected. One can also try to identify all batches in one solution which corresponds to solving a minimum set covering problem. Hirabayashi et al. [1984] give such a formulation and suggest a column generation procedure for its solution. The

subproblems for generating the columns are the optimal tool module design problems. However, the problem is still NP-complete.

A formulation to cover the production orders set by partitioning it into batches is also suggested in Rajagopalan [1986]. A detailed discussion of his formulation is given in Sec.2.3.2 within the context of machine loading. Due to the NP-complete nature of the problem, two groups of heuristics are designed for its solution. In these heuristics, the problem is treated as an m -dimensional bin packing problem and adaptations of the First or Next Fit rule are applied. The objective is the minimization of the total completion time. In the first group of heuristics, parts are assigned priorities with respect to their tool slot demands. Higher demands are assigned higher priorities. In the second group, parts are ordered considering their processing time requirements. The second group of heuristics does significantly better than the first in terms of completion times.

Partitioning production orders into groups so that the total machine idle time for all groups is minimized is the objective defined for part-type selection in Afentakis, Solomon, and Millen [1989]. A cyclical scheduling policy is assumed in which the total processing time on the bottleneck machine determines the cycle time for each group. Only one unit from each part-type is processed in each cycle, so that the lot size which needs to be computed externally will determine the number of cycles. An MIP formulation is given for which bin packing heuristics are proposed. Multifit heuristic is used for total idle time minimization and also for the minimization of the number of families along with the total idle time. Based on the proposition that partitioning parts

into more families never improves the total idle time, a second heuristic called "Contraction" is suggested. This heuristic starts with each part-type as a separate group and combines them based on tool magazine limitations until no further contraction is possible. Multifit yields better solutions.

Another bin packing heuristic, First Fit Decreasing, is used in Tang and Denardo [1988b] to find an upper bound on the optimal value of the objective function which is defined as the total number of instants at which tools are switched. The solution gives the batches of part-types. A branch and bound procedure that enumerates the sequential maximal partitions of the production orders set is developed. In order to find a lower bound a compatibility matrix is constructed where two part-types are shown to be compatible if their total tool slot requirements do not exceed the magazine capacity. The partitioning procedure selects a part-type which is compatible with the least number of part-types and groups them together. When the production orders set is covered, an approximate solution is obtained to the optimal parts grouping problem as defined in Hirabayashi et al. [1984].

The part-type selection problem can also be approached from the perspective of Group Technology (GT). Kusiak [1984,1985c] discusses various classification and coding schemes applied in GT and proposes a pattern recognition methodology for parts grouping. He also describes a selective grouping approach which is based on certain attribute values of part-types in reference to the FMS structure such as the required fixture, pallet, and tool types. In order to formulate and solve the parts grouping problem, clustering analysis is applied. A p-median formulation and a matrix

formulation of the problem are suggested. In the p-median formulation, p groups are formed such that the total distance of parts from each other is minimized. The distance between any two parts is computed based on their attribute values and is a measure of their similarity. In the matrix formulation, the problem is presented in the form of a matrix whose rows are the part-types and columns are the associated attribute values. By rearranging rows and columns in this matrix, part groups are generated. There are algorithms for making these rearrangements. Kusiak and Chow [1987] develops a clustering algorithm which is polynomial in its computational complexity and easy to implement. However his GT based approaches to the part-type selection problem do not consider the system constraints such as the limited number of pallets and fixtures or the limitations on the tool magazine capacities.

While forming part groups other FMS components such as machines, tools, pallets, and fixtures can also be grouped so that each part-component group will be treated separately. This will decompose the FMS into ideally independent subsystems and will ease the planning and scheduling decisions. The problem of grouping parts and machines using this GT concept is dealt with in Kumar, Kusiak, and Vannelli [1986]. Part machine relations are represented by a bipartite graph and an optimal k-decomposition formulation is given in which the graph is decomposed into k subgraphs such that sum of the weights on the edges between the subgraphs is minimized. The number of subgraphs, k, corresponds to the number of groups and is a control parameter. This is a quadratic assignment formulation which is known to be NP-complete. Kumar et al. develop a two phase polynomially bounded heuristic for its solution.

Chakravarty and Shtub [1984] suggest a similar idea for grouping parts and tools. They propose to use clustering algorithms. Tool magazine capacity and the number of available pallets and fixtures are considered as limitations on the number of parts and tools in each group.

A drawback of these GT approaches is that in real manufacturing environments an ideal partitioning of parts and components is almost impossible. By ideal partitioning we mean forming totally independent groups such that there is no sharing of components (machines, tools, pallets or fixtures) by the parts in different groups. Since this cannot be achieved, there will either be duplications of such components in different groups in which they are needed or some parts will be allowed to move between groups.

A different approach to batching is presented in Whitney and Gaul [1985] which involves a sequential decision criterion. A performance index which denotes the probability of successful outcome as a function of all part-types remaining to be batched is determined. Success is completing the decision process without violating a system constraint. Each part-type is assigned an individual performance index based on its contribution to system's performance with respect to factors such as machine utilization, degree of tool sharing, due dates, or tool magazine capacities. These performance indices are probabilistic measures and a higher probability indicates a higher contribution to success. Within a probabilistic framework, this sequential procedure handles the part-type selection problem comprehensively. In addition to part-type selection, machine loading according to the selected part-types is also a part of the procedure.

Flexible approach: An alternative to the batching approaches is proposed by Stecke and Kim [1986]. In this approach new part-types are introduced into the system when a part completes its processing and hence frees some slots on the tool magazine. Their formulation combines machine loading and production ratio determination with part-type selection. The objective is to minimize a weighted sum of deviations from an aggregate workload on each machine, while selecting the part-types and their required tools and determining their relative ratios. These ratios are constrained by fixture limitations. However, there is no consideration of tool magazine capacities in this formulation. A simulation study is performed to test the suggested flexible algorithm. The results show that the flexible approach provides higher utilizations and lower flow times when compared to a batching approach. This result is also confirmed in Stecke and Kim [1988] where the flexible approach is compared to the batching approaches of Hwang [1986], Rajagopalan [1986], and Whitney and Suri [1984]. In this work, the model used in the flexible algorithm incorporates the tool magazine restrictions. The flexible algorithm is implemented iteratively until all parts complete their requirements. As in their previous work, simulation is used to determine the completion patterns. When a part-type completes its requirements, it is deleted from the model; the parts still being processed are forced to have positive ratios; and the model is solved again to give the new part-types and their ratios. The results indicate that Hwang's model with an extension for weighing the part-types with respect to their tool slot demands in the objective function gives the smallest number of batches among all the approaches. The flexible approach appears to be better than the others in terms of system utilization.

2.2.4. Machine Loading Problem

Problem definition: Machine loading is the assignment of operations of the selected part-types together with the required cutting tools to machines or machine groups. Being the last one of the five planning problems as defined by Stecké [1983], it assumes that part-types, machine groups, production ratios, pallet and fixture allocations have already been determined. The purpose is to make operation assignments in such a way that tool magazine capacities are not violated and production is maximized. However as mentioned earlier, machine loading may also be formulated together with other planning problems. Although an extensive formulation which deals with all the planning problems may not be computationally feasible, to deal with some combination of subproblems may provide more insight and better results. In this section we review various approaches to the problem which either treat machine loading separately or combine it with other planning issues.

There are alternative operational objectives that serve the ultimate goal of production maximization. Stecké [1983] identifies six loading objectives which are valid for different operational settings. The most widely used objective is balancing the workload per machine. It is known that a balanced workload will increase the system throughput by eliminating unnecessary congestions. However Stecké and Solberg [1985] show that for a system of unequally sized machine groups, expected production rate is maximized by assigning a specific unbalanced workload per machine to each group. Their results are based on a CQN analysis. Balancing the workload per machine is optimal only if the group sizes are equal. Stecké and Morin [1985] analyze the

optimality of balancing in systems where there is a fixed routing of parts and no machine grouping. They too use a CQN model of an FMS and exploit certain concavity properties of the production function to establish the global optimality of balancing. Based on these results, balancing workload per machine groups of equal sizes and unbalancing the workload per machine groups of unequal sizes are identified as two loading objectives.

Another objective for machine loading which may be conflicting with workload balancing is the minimization of parts' movements among machines. In a simulation study conducted on a real FMS for testing various loading policies Stecké and Solberg [1981] surprisingly report that a parts' movements minimization policy gives almost the best results in terms of average number of completed part-types although the resulting workloads are extremely unbalanced.

Other loading objectives identified in Stecké [1983] are filling the tool magazines as densely as possible and maximizing the sum of operation priorities. These two objectives will help to increase the routing flexibility of the system by duplicating tool and operation assignments. Particularly if weights are assigned to operations in such a manner that the bottleneck operations are allocated to more than one machine, waiting times will decrease and production will increase.

Stecké and Talbot [1983] present heuristic solution procedures for the FMS loading problem, with different objectives such as minimizing part movement and balancing and unbalancing the workload.

Another relevant objective for the machine loading problem is suggested as the minimization of production costs. Different machine efficiencies or varying efficiencies of tools on different machines cause different processing costs. Such cost related loading criteria are referred to in Kusiak [1985a], Chakravarty and Shtub [1984], Sarin and Chen [1987].

In another line of research where due dates are considered during machine loading or when the loading problem is treated in conjunction with scheduling decisions, criteria like makespan minimization may be appropriate. Rajagopalan [1986] formulates such an objective function.

Review of various approaches: With the selection of an appropriate loading objective the loading problem can be formulated as a mathematical program. The most significant system constraint is the tool magazine capacity. Therefore, in all of the following papers discussed, consideration of tool-operation relations and tool magazine limitations is essential. Machine time availabilities and availability of other system resources such as pallets and fixtures are also the basic elements which should be incorporated.

Stecke [1983] formulates the problem as a binary program for various loading objectives. These formulations involve nonlinearities both in their objective functions and in the constraint set. Nonlinearities in the constraints arise from the need to consider tool slot savings because of tool sharing and physical placement of tools on the tool magazines. In the solution procedure developed, the models are linearized to result in 0-1 MIPs. However, these linearizations result in considerably a larger number of constraints and the resulting formulations take considerable computing effort. A direct

solution procedure is developed in Berrada and Stecke [1986]. The loading objective is to balance the assigned workload on each machine. The solution procedure consists of a sequence of subproblems, each defined by fixing the maximum workload, whose solutions converge to an ϵ -optimal solution.

Using the branch and bound algorithm in Berrada and Stecke [1986], Kim and Yano [1987b] try several measures of workload balance or unbalance to maximize system throughput. The only difference is in the objective functions which are defined separately for each measure. For workload unbalancing a slight modification is made in the problem formulation. For workload balancing the objective which minimizes the maximum workload yields the best results in terms of system throughput. For the unbalancing case, the objective which minimizes the maximum ratio of overload to the ideal load among the machine groups performs best. Kim and Yano [1989a], develop an improved branch and bound approach for the same problem. The new algorithm is faster and provides better quality solutions.

In these formulations of the loading problem there is no due date consideration for the part-types. It is assumed that due dates had been accounted for during the part-type selection stage. A similar formulation of the FMS loading problem with an additional criterion of meeting the due dates, is given in Shanker and Tzen [1985]. Two alternative objective functions are considered. One is the workload balancing objective and the other is a bi-criteria objective which maximizes the weighted sum of the number of jobs selected while minimizing the unbalances over all machines. The model is solved in each scheduling period and the jobs which have due dates scheduled for the next two

periods have higher priorities in the objective function. Heuristic approaches are suggested. The same model with the objective of maximizing the assigned workload is solved with a branch and bound procedure in Shanker and Srinivasulu [1989]. Since tool slot savings are not considered there is no nonlinearity involved. Still, the computational burden of the solution procedure necessitates the development of heuristics. These heuristics try to achieve the objectives of balancing the workload and maximizing the throughput simultaneously.

Similar to those in Shanker and Tzen [1985], two sets of heuristics, one for the objective of workload balancing and the other for the bi-criteria objective of meeting due dates and balancing workload are developed in Moreno and Ding [1989]. These heuristics involve the evaluation of the balancing objective in the selection of part-types. The part-type and the associated route that satisfies the tool magazine constraint and contributes the most to the improvement of the objective are selected at each iteration. Results are reported to be better than those obtained by Shanker and Tzen's heuristics.

Another bi-criteria approach to the loading problem is presented in Ammons, Lofgren, and McGinnis [1985] for a flexible assembly system (FAS). Although the assembly system characteristics are much different, the assignment of components and assemblies to work stations in an FAS is similar to the assignment of tools and operations to machines in an FMS. The bi-criteria objective function to be minimized is a weighted sum of the maximum deviation from the ideal workload over all work stations and the total number of job movements among work stations. The model is a large scale BP which is difficult to solve even when some constraints are relaxed. Heuristic procedures

based on constrained longest processing time (CLPT) rule and clustering techniques, are developed.

Minimization of parts movements is chosen to be the loading objective also in Shanker and Rajamarthandan [1989]. The model is similar to the one formulated by Stecke [1983b] to minimize the number of movements between the machines. There is an additional constraint for tool copy availability. Hence the model has a similar computational burden.

The models discussed so far do not deal with refixturing between operations explicitly. An extension of these models to include refixturing and tool availability is given in Lashkari, Dutta, and Padhye [1987]. Two alternative objective functions are formulated to minimize the transport load on carts and to minimize the number of refixturings required when consecutive operations are performed on different machines. The resulting model is a binary nonlinear program and requires considerable computational effort. The same problem is reconsidered in Wilson [1989] where a simpler formulation is suggested which removes the nonlinearities due to tool sharing and limited tool availabilities. The resulting model is solved using a standard code and solutions are obtained quite efficiently.

A loading objective different from those discussed so far is the minimization of the total processing costs. Kusiak [1985a] gives such a formulation considering different machine efficiencies in the processing of operations. The suggested IP model avoids nonlinearities in the tool magazine constraints at the expense of reducing machine capabilities due to tool duplications. Another feature included in the model is the

consideration of tool lives. A simplification in the model is proposed by allocating batches as opposed to operations. This is claimed to be consistent with the idea that FMSs are designed to handle a wide variety of part-types in small batches. However, this assumes continuous divisibility of batches. The solution procedure suggested for the linear IP is based on subgradient optimization. Different machine efficiencies are also accounted for in Chakravarty and Shtub [1984]. In their model, tools are assigned to machines in such a manner that maximum processing time over all machines are minimized. Each tool is assumed to have a different efficiency and therefore a different processing time on each machine.

Minimizing the total machining costs where tools have different efficiencies on different machines is the objective in Sarin and Chen [1987]. The suggested model includes many aspects of the loading problem such as tool life, tool magazine capacity, machine time capacity, and system congestions. The nonlinearities due to tool sharing are removed by defining two sets of variables, one for operation assignments and one for tool assignments. Defining variables in this manner also avoids unnecessary tool duplications. The model grows too large to be computationally feasible even for small problems. However, through deleting the infeasible tool-machine and tool-operation assignments, considerable size reductions may be achieved. Still, direct solution techniques may not be computationally feasible. Lagrangian relaxation approach with alternative relaxations is proposed.

Kouvelis and Lee [1991] presents a loading model with the objective of minimizing tool operating costs. Operations are assigned to machines according to the

availability of tool magazine and time availability constraints. By defining new variables, a model with a block angular structure is obtained. Exploiting this structure, a decomposition based branch and bound algorithm is developed. This approach provides considerable improvements when compared to a general branch and bound method.

In the formulations of the FMS loading problem, it is assumed that part-types have been selected, presumably based on their due dates, and their production ratios have been determined. If due dates of the part-types are to be considered in the loading problem, then the approach of combining it with part-type selection will be appropriate. That is, only the operations of the part-types which are being selected will be allocated to machines in that scheduling period, leaving the unselected part-types to the next period. This is the approach in Shanker and Tzen [1985] as discussed previously.

A formulation which combines three planning problems is presented in Rajagopalan [1986]. The MIP model constructed links the problems of part-type selection, production ratio determination, and loading. A distinguishing feature of his formulation is the consideration of the entire planning horizon by dividing it into tool set-up periods. In each period a new subset of the production orders is processed which may require a major tool changeover before production starts. The objective is to minimize the overall makespan by minimizing the sum of the makespans of individual set-up periods and the total set-up times due to tool changeovers between each period. A constant set-up time is assumed. The number of set-ups is a variable which depends on the completion of all production orders. Different from some of the loading formulations discussed above, loading is determined by assigning part-types rather than operations, to

machines. This requires the assumption that a part-type either has only one operation on a machine or all of its operations on a certain machine type are assigned to the same machine. Solving the resulting MIP model to optimality is computationally infeasible from a practical point of view. Hence, Rajagopalan develops bin packing type heuristics as discussed in Section 2.2.3. The solution obtained from these heuristics provides batches of part-types to be processed sequentially and their corresponding tool assignments. However, the production ratios at which these part-types are processed, which was one of the decisions in the MIP formulation, cannot be determined.

Bin packing type heuristics for solving the loading problem are also developed in Kim and Yano [1987a]. The objective is to maximize throughput via balancing or unbalancing workloads. In the two dimensional bin packing formulation the tool magazine capacities and the machine time availabilities are considered. The items (operations) are assigned to the bins in such a way that their widths (number of tool slots on the magazines) are not exceeded while sum of the heights of the items (processing times) is close to the ideal heights of the bins (machine capacities). The ideal heights are the parameters which can be computed in an aggregate and theoretical manner using the CQN approach of Stecké and Solberg [1985]. Tool savings are considered by using a labeling method. The suggested procedures are mainly based on LPT and multifit heuristics and their combinations.

The flexible approach of Stecké and Kim [1986,1988] discussed in Section 2.2.3 suggests an alternative to introducing new part-types into the system. In this approach, part-type selection, production ratio determination, and machine loading decisions are

incorporated in one formulation. Stecké and Kim [1989] use this approach to investigate the effect of unbalanced aggregate workloads for machine groups of unequal sizes. They make a comparison of balancing and unbalancing workloads in an existing FMS. The results indicate that unbalancing leads to higher utilizations in terms of machines, transportation and blocking and also to shorter makespans.

A shortcoming in most of the models discussed is the lack of attention paid to due dates. When the loading problem is dealt with alone, it may be assumed that due dates had been taken care of in the previous planning phases. However, when loading decisions are combined with part-type selection and production ratio determination, due date considerations become very important.

In Bastos [1988], whose hierarchic approach to FMS planning is discussed in Section 2.2.2, due dates are taken into account by specifying lower bounds on the production amounts. In the batching model, which is formulated as an LP, the number of parts of each type to be assigned to a route in the next period is determined while maximizing the weighted number of parts. There are machine time and tool magazine availability constraints. However, tool sharing is not explicitly considered. In the iterative solution procedure, tool sharing is considered by updating the related model parameters at each iteration. A separate algorithm computes the lower bounds on the production amounts based on their specified due dates. A routing model assigns a route to each part-type while minimizing the makespan and balancing the workload among nonbottleneck machines iteratively.

Chung and Doong [1989] also approach the problem by combining the part-type selection and machine loading decisions with due date considerations. First, an optimal routing mix is determined for each part-type by solving an LP as if only one part-type will be processed in the next scheduling period. Then based on the capacity required from the bottleneck machine, a critical ratio is computed for each part-type to check the feasibility of meeting its due date. Part-types whose due dates cannot be met are discarded. Then a check is made to see whether the due dates of the remaining part-types can be met. Once a feasible part-type mix is found, an MIP model is solved with the objective of makespan minimization to determine the tool assignments and production ratios. The results are tested by simulation and if needed a re-planning is implemented with updated values of certain critical parameters. However one would expect such an iterative procedure, which involves solving a fairly large scale MIP and uses simulation as a feedback mechanism, to have unwieldy computational requirements. The authors suggest using search methods to find near-optimal solutions. The solution procedure suggested in this paper actually includes another planning decision; determination of the amounts at which the part-types will be produced along with their routing mix. Loading formulations in the previous papers assume predetermined production ratios and fixed routing of parts except for the cases in which operations of a given part-type are allowed to be assigned to more than one machine. This, however, does not determine a routing for parts but instead it provides a flexibility in routing which eases the real time scheduling decisions.

Shirley and Jaikumar [1989] presents a mixed integer nonlinear model for flexible transfer lines (FTL) in order to determine the parts to be manufactured on the line and the lot sizes and batches in which they should be produced. Their objective is to maximize system utilization. In the solution procedure, the model is linearized and then some of the constraints are relaxed along with the integrality constraints. This allows for a decomposition where each batch can be treated separately. They state that the approach quickly converges to an optimal or near optimal solution.

Lot sizing in a flexible assembly system (FAS) is addressed in Chang and Sullivan [1986]. An MIP model which minimizes a cost function subject to demand and capacity constraints is developed. The capacity constraints include the set-up time for changing tools and set-up activities between consecutive batches. The model can be solved optimally as long as the number of part-types is not too large.

O'Grady and Menon [1987] point out the importance of due dates in their goal programming formulation of the loading problem. They consider an existing FMS in Scotland containing six CNC machines. This system is described in Carrie, Adhami, Stephens, and Murdoch [1984] within the framework of introducing an FMS in a company. The part-types are heavy castings and each unit is treated individually in the model which selects parts from the candidate order population to be released for processing in that planning period. The objective is to minimize a weighted sum of deviational variables which are defined to monitor over-achievement or under-achievement of preset targets. These are due dates, machine utilizations, tool magazine loads, and order priorities. The model also considers tool copy availabilities and handles

tool sharing without any nonlinearity. Six alternative schemes are identified for weighing the alternative criteria in the objective function. Using a standard code, the solution of the model required an average of 25 minutes. The solutions obtained are not global optima but compromise solutions which proved to be satisfactory in incorporating different preferences.

The same FMS is also referred to in Carrie and Perara [1986]. They consider the tool loading problem in relation to input sequencing decisions. In this particular FMS, the number of tool slots occupied by tools differ considerably with respect to their sizes and shapes, and tool magazine capacities fall behind the needs. Due to the system characteristics, determination of a part input sequence as suggested by Carrie and Petsopoulos [1985], to minimize the number of tool changeovers does not result in any significant improvement in the system performance. It is also observed that fixture limitations have a more significant impact on the release sequence of parts rather than some other priority rules. Dynamic selection of parts from the queue to minimize the tool changeovers does not appear to be effective either. An important observation made is that the number of tool changes due to tool wear is larger than those due to product variety. A different approach to FMS loading is addressed in Han, Na, and Hogg [1989], with the idea of moving tools instead of parts movements among machines. When a part requires a tool which is not available on the tool magazine, it is delayed on the machine until the required tool is borrowed from another machine. A nonlinear IP model is constructed to minimize the total number of tools borrowed over all machines. This model also incorporates tool copy availability limitations. An approximation

procedure is suggested where the problem is divided into two subproblems, one in tool variables and one in part variables. The subproblem in tool variables is a capacitated transportation problem and the subproblem in part variables has the form of a generalized assignment problem. As an alternative approach, a greedy heuristic based on the LPT rule is designed to assign parts to machines and then to assign tools to machines where they are required most often. In order to implement the loading results, various operating policies, such as immediate tool return, no tool return, dedicated queues, common queue, along with various dispatching rules are suggested. These are tested by simulation experiments using throughput as the performance measure. No significant differences are observed with respect to different dispatching rules.

In these last two papers and in some others which evaluate the loading decisions in combination with certain dispatching rules input sequencing of parts are also considered. In the next section we review more of the work related with sequencing and scheduling decisions.

2.2.5. Other Planning Problems

FMS planning also involves decisions on grouping machines, determining relative ratios of part-types to be processed simultaneously on the system at any time, and allocation of other FMS components such as pallets and fixtures to part-types. Usually these decision are incorporated in part-type selection and/or machine loading problems. Most of the loading models discussed previously include constraints to account for pallet and fixture availability. Production ratios can also be determined while allocating operations to machines as in Rajagopalan [1986], Stecke and Kim [1988]. Machines can

be grouped while selecting part-types using GT approaches as in Kumar, Kusiak, and Vanelli [1986], Chakravarty and Shtub [1984]. As discussed previously in Section 2.2.4, Stecke and Solberg [1985] consider the partitioning of machines into groups using a CQN model.

Stecke [1986] refers to the machine grouping problem in more detail considering the tool magazine capacity restrictions. The procedure consists of two steps. First the minimum required number of machine groups is determined by solving a nonlinear IP model which considers tool slot savings due to tool sharing and physical placement of tools on the magazines. In the second step optimal pooling is determined based on the results obtained in Stecke and Solberg [1985]. However, the nonlinearities involved in the formulation in the first step makes the problem difficult to solve. If this problem is to be solved at the beginning of each planning period which is usually a day or a shift, more easily implementable methods should be developed. Also as mentioned in Rajagopalan [1986], forming machine groups which are identically tooled decrease the flexibility of performing diversified operations.

Stecke [1985] develops procedures in order to determine production ratios to maintain balanced or unbalanced workloads on machines. LP and IP models are formulated which minimize a weighted sum of deviations from the ideal aggregate workloads. The results show that the production ratios obtained at each IP iteration are quite different with very close objective function values. Stecke makes a note that this may suggest the use of a secondary criterion such as due dates to select the best ratios. However, this requires a more detailed consideration of transportation times, queuing,

and expected down times. A simulation study which analyzes various scenarios to determine the minimum inventory levels to achieve the machine utilizations provided by aggregate production ratios is presented in Schriber and Stecke [1986]. Various factors, such as the number of automated guided vehicles (AGV), transportation times, number of buffers available are considered and sensitivity of theoretical machine utilizations suggested by aggregate ratios to these factors is measured. The results show that machine utilizations and production rates depend more on the level of work in process (WIP) than on the number of AGVs and that there are WIP levels and a number of AGVs which give machine utilizations and production ratios close to those suggested by aggregate conditions.

2.3. FMS Scheduling

Problem definition: Once the FMS has been set-up for production in the next planning period, problems related to the real time running of the system arise. These involve decisions about input sequencing of part-types, sequencing the operations of part-types, selection of machines to perform the operation, and selection of the part-type to be processed next on a machine. Machine loading decisions have a direct effect on FMS scheduling. Ideally these two problems should be considered together. However, this may not be feasible for two reasons. First, since most FMSs are not equipped with automated tool handling devices, it is not efficient to make frequent tool switches. Hence, scheduling of parts are based on the tooling decisions that have already been made. Second, if these two decisions are to be made simultaneously, then the problem

becomes more complicated to deal with. There are studies in the literature where authors evaluate the performance of various loading objectives in combination with certain dispatching rules.

Stecke and Solberg [1981] compares five loading policies under sixteen dispatching rules for real time flow control by simulating a dedicated FMS. Some of the priority rules tested are SPT, LPT, FOPR (fewest operations remaining of each part), MOPT (most operations remaining), SPT/TOT (smallest value of processing time for the operation divided by the total processing time for the job). The loading policy with the objective of minimizing parts movements gives good results in terms of the average number of completed parts. Best results are obtained by pooling machines. Among the dispatching rules SPT/TOT provides the best results. Investigation of loading policies in conjunction with certain dispatching rules is also undertaken in Shanker and Tzen [1985] whose loading approach was discussed in Section 2.2.4. They perform a simulation study to test five loading policies with FIFO, SPT, LPT dispatching rules. Stoeva [1990] develops a due date based dispatching rule in a cellular manufacturing environment.

Other studies concerning FMS scheduling usually assume that machine loading decisions have been made and determine the input sequencing of parts considering more detailed issues such as precedence relations, WIP buffers, and transportation times. Performance criteria may be makespan, mean flow time or tardiness. Scheduling policies differ with respect to the characteristics of the flexible system referred. In systems where certain production ratios of parts are maintained and part variety does not often change, periodic or cyclic policies become appropriate. In more random FMSs dynamic

scheduling policies are required. A classification of FMS scheduling environments and a review of related literature is given in Rachamadugu and Stecke [1989]. Our concern with scheduling is a result of our desire to establish its links with machine loading. Hence, we look at the literature from the viewpoint of either how these two decisions can be integrated or rather how might the loading problem be formulated in order to ease the subsequent scheduling decisions.

Review of various approaches: A combined consideration of the FMS loading and scheduling decisions is presented in Afentakis [1986]. However, the loading problem formulated to minimize the cycle time does not include the tool magazine restrictions. The environment is a dedicated FMS where part-types are produced according to a master production schedule (MPS) to feed an assembly line. Cycle time is defined as the time required to produce one unit of each part-type with the assumption that MPS comprises of one unit from each. The model considers alternative routings of parts and the precedence relations of operations. A heuristic procedure based on a graphical representation, which is then enumerated to find a maximal independent set, is proposed. Afentakis also proposes a procedure in which the transportation times are considered for the purpose of minimizing lead times and WIP inventories. For sequencing of operations on each machine a job-shop makespan minimization formulation finds the best sequence. Afentakis, Solomon, and Millen [1989] develop a heuristic procedure for minimizing the WIP inventories in FMSs which use cyclic schedules. These two studies are based on the assumption that machines are loaded with the required tools and that there is a feasible route for each part-type. Although the first study considers routing, it does not account for tool assignments.

A similar FMS environment is referred to in Erscler, Roubellat, and Thuriot [1985]. Each part-type has to be processed in a predetermined ratio over a certain planning horizon. That is, the problem of production ratio determination has been solved and this minimal part set will be released periodically as proposed by Hitz [1979,1980] for flexible flow shops. For each part-type which has been chosen at the planning level, there is a fixed routing. The releasing period has two lower bounds, one coming from the maximum total processing times over all machines and the other is from the pallet restrictions. Also, there is an upper bound on the flow time of each part-type in the system due to pallet availability. These maximum flow times and the length of the releasing period are inputs in the determination of limit times which are the earliest start and the latest finish in each periodic schedule. These limit times are determined considering the maximum part flow times, periodic steady state conditions, and precedence relations of operations. The procedure is iterative and is based on resolving conflicts due to infeasible time overlaps, by updating the limit times.

A two phase algorithm for real time scheduling is developed for random FMSs in Chang, Sullivan, Bagchi, and Wilson [1985]. It is an approximation method which considers the changes in the system status dynamically as operations are completed and new jobs arrive. A new job arrival triggers rescheduling the system. However, there is no consideration of tool requirements. It should be assumed that minor tool switches are made as required upon the completion of an operation as in the flexible approach of Stecke and Kim [1986,1988]. The first phase of the method generates many feasible schedules and the hope is that this set of feasible schedules will include an optimal or

near optimal schedule. This generation scheme is based on work station availability which is constrained by previously scheduled jobs. The second phase uses an optimization procedure to select a schedule for each job in order to minimize total completion times. It is a binary programming model solved by a branch and filter algorithm which is exponential in the number of feasible schedules and machines. Simulation experiments are designed to evaluate the method's performance by comparing it against dispatching rules such as SPT, LPT, MWKR, LWKR. Other factors such as utilization of the bottleneck machine, information usage (look ahead policy), and batch sizes are also considered to investigate their effects on system's performance. A comprehensive statistical analysis of the results leads to the conclusion that the proposed method performs better than the dispatching heuristics. Look ahead policies always proved to be better. Batch sizes do not affect the results. This is expected since there is no consideration of set-up times.

Dynamic scheduling of parts is also studied in Birge [1985]. It is assumed that a pre-schedule to consider set-up and inventory decisions up to a certain planning horizon has been developed at a higher decision level. The purpose is to deal with system disruptions in real time to minimize tardiness related costs.

Maimon and Choong [1987] address dynamic scheduling issues in a reentrant FMS. Main emphasis is to maintain the balanced workloads on the identical parallel work stations against stochastic disruptions. This is achieved by transferring WIP inventories across work stations when necessary. A model for this WIP flow is presented in which a convex function of the sum of expected transportation, backlogging,

and inventory costs is minimized. A stationary policy is determined by using a dynamic approach. Performance results obtained via simulation show improvements in terms of throughput, workload balance, and reduction of WIP.

The problem of finding an optimal scheduling scheme in a random FMS operating within a static environment is studied in Hutchinson, Leong, Synder, and Ward [1989]. The formulation considers the loading and scheduling problems simultaneously with the objective of makespan minimization. There are alternative routings for jobs and, it is assumed that tool magazine capacities are large enough not to constrain the system. A mixed BP model is formulated which considers the precedence relations among operations and assigns each operation to only one machine. By relaxing the precedence constraints a lower bound on the original objective function is computed. This optimal scheduling scheme is compared to a decomposed scheme and a look-ahead control policy using SPT rule. The decomposed and optimal schemes give highly increased performances over the look-ahead policy justifying the additional computational effort. However, the increase in performance provided by the optimal scheme over the decomposed scheme does not justify the additional computational effort.

Makespan minimization is the scheduling objective in the mixed BP formulation of Greene and Sadowski [1986]. The difference is that they consider an FMS comprising of flexible manufacturing cells (FMC) of two or more machines for which tooling decisions have already been made. Hence each job is first assigned to an FMC that contains all machine types capable of performing all of its operations. Jobs are then scheduled within the FMC. The model is intractable for problems of reasonable size.

Another makespan minimization based scheduling approach is presented in Sherali, Sarin, and Desai [1990]. Each job has an associated "alternate routing combination" (ARC) list which specifies the various ways of performing the job. The BP model selects a subset of jobs along with an ACR while minimizing the makespan. They design enumeration based heuristics and a Lagrangian relaxation heuristic where the Lagrangian problems are solved using a dynamic programming approach. The Lagrangian relaxation heuristic yields strong bounds and good solutions but is not computationally efficient for real size problems.

Sequential consideration of machine loading and scheduling problems is studied as a part of a four level hierarchical structure in Sawik [1990]. During the loading phase, operations of the selected part-types are assigned to machines to be processed at their required amounts. Lot splitting for a part-type is allowed. Although tool magazine restrictions are considered, there is no machine capacity and precedence relations at this level. Hence the output of the loading model will provide a lower bound on the actual completion time. Operations scheduling problem is also formulated as an IP with the objective of minimizing the maximum lateness. Based on the output obtained from loading, a period by period scheduling of operations is considered. Load/unload operations are also accounted for and new part-types are released into the system according to pallet availability. An approximation algorithm is devised to solve the model. It is a period by period assignment procedure, which assigns operations to machines in the order of job urgency. A job's urgency is defined as the ratio of the longest processing time among different routes to its due date. Another algorithm, which

basically uses the same selection rule, is devised for the initial sequence of the parts. The lower bounds obtained from the loading problem are used to evaluate the performance of the scheduling heuristics. The average deviations are 21 % of the lower bound on C_{\max} (minimum makespan) and 106% of the lower bound on L_{\max} (maximum tardiness). The most significant contribution of this study is in its attempt to integrate the planning and scheduling subproblems and also the consideration of due date based decision criteria.

Minimizing the maximum tardiness is also considered in Pourbabai [1986] within the context of applying just-in-time manufacturing (JIT) and GT concepts to an FMS. There are jobs of homogeneous part-types with predetermined availability times and due dates. End product inventories are not allowed. Work stations are designed based on GT and all operations of a job in a particular work station are aggregated into one. A mixed BP model is formulated based on prespecified sequencing/ dispatching rules. Jobs with earlier due dates are dispatched first and among jobs with equal due dates the one with smaller availability time is dispatched first. Otherwise the sequence is arbitrary. A standard code is used for its solution, however no computational results are reported.

Another due date based scheduling problem in a general FMS environment is presented in Raman, Talbot, and Rachamadugu [1989]. Job arrivals are random and the objective is to minimize mean tardiness. This is a dynamic scheduling approach as in Chang, Sullivan, Bagchi, and Wilson [1985]. When there is a change in the system status, operation due dates are generated from job due dates in a dynamic manner based on the new state of the system. Raman et al. also investigate the impact of unequal

machine workloads on the effectiveness of the due date based dispatching rules. As an alternative approach, a static problem is formulated which needs to be solved on a rolling basis at every instance the system status changes. Although it is a large scale integer program, the authors claim that it can be solved with "reasonable" computational effort due to its resource constrained project scheduling problem structure. The model also considers transportation times by treating the transporter as a resource. The results of the simulation experiments show that the selection rule with respect to machine workloads improves system performance. Best results in terms of mean tardiness is obtained by implementing the near optimal solution of the static problem on a rolling basis.

A flow control methodology based on the objective of meeting time varying demands for part-types in a multi stage FMS, each stage having parallel machines is given in Han and McGinnis [1989]. Flow is unidirectional through the stages and bypassing is allowed. Within a stage options are available with machines that have already been loaded with the required tools. There is a limited buffer between stages and input and output buffers are also available. The planning horizon is divided into control periods and the amount of flow through each stage in a control period is determined so as to minimize the stockout costs. Different optimization models are constructed to handle different system states with respect to the feasibility of the projected demand, projected minimum flow, and the starting output inventory. In the beginning of each period an appropriate model needs to be solved. The release, dispatching, and routing decisions are based on simple rules. A simulation experiment is designed to test the

suggested methodology and the results appear encouraging. A nice property of the method is that the problem size is not a restriction and it is simple and easy to adapt it to changes.

Kimemia and Gershwin [1985] develop a network flow optimization model to find the optimal parts routing in a FMS. The objective function is defined to maximize a performance measure such as throughput. The model determines the flow rate of parts to machines and part production rates. The problem is solved using a decomposition approach. Optimal assignments are then tested via simulation. The resulting values of the performance measures are close to their optimal values.

The various scheduling approaches discussed so far consider either dedicated or random FMSs. Akella, Choong, and Gershwin [1985] study a real FFS, the IBM cardline assembly line. They apply a hierarchical control mechanism to cope with disruptions such as machine failures or deviations in demand. The top level in the hierarchy computes the hedging points for each part-type as a function of the system status. Production rate for each part-type is determined in the middle level with respect to the system status and the accumulated buffers. When the buffer level reaches the hedging point the production rate will be equal to the demand rate. This is formulated as a cost minimization linear program in which the cost coefficients involve the differences between buffer levels and hedging points. The lower level dispatches the part-types based on the projected buffer levels. A part-type is loaded when the actual production supply is less than its projected value. This policy is tested against three other policies which are not based on capacity considerations. Simulation experiments

are made with the objective of meeting the production requirements of part-types by the end of one shift. The results indicate that the hierarchical policy gives higher production percentages and lower WIP inventory levels and also provides a better balance of production among part-types.

Tang and Denardo [1988a] presents a mathematical model for job scheduling for one flexible machine. They try to find a sequence of jobs that will minimize the number of tool switches. Since the model is computationally intractable, they design a heuristic procedure to find a local optimum solution to the problem.

Kusiak [1986b] considers a FMS consisting of both machining and assembly operations. Machining and assembly subsystems are linked by an automated guided vehicle (AVG). Parts are first machined and then sent for assembly. The scheduling problem is formulated at two levels. At an aggregated level a two machine flow-shop problem is solved. For the detailed problem product and part priorities are determined. A job schedulability status vector is defined considering factors such as precedence relations, pallet/fixture limitations, and tool availability. By updating an active jobs list and a waiting list in an iterative manner jobs are scheduled on machines.

2.4. Summary

Research on FMSs concentrate on many different issues starting from economic justification and design problems to planning, scheduling, and control problems. Due to the flexibilities they provide, a sound management of these systems involve solving problems most of which are more complicated than those confronted in conventional manufacturing. OR techniques and methods have been extensively applied to deal with

many of these problems. Queuing theory and simulation techniques are mostly used to answer design related questions. Since all of the features of an FMS cannot be incorporated in queuing models, their results provide highly aggregated values of the specified performance measures. Simulation studies are performed to investigate the effects of these aggregated results in more detailed operational settings. For FMS planning problems, mathematical programming has been widely used as a modelling technique. Linear/nonlinear integer, mixed integer or binary programs have been formulated to solve part-type selection, resource allocation, machine loading, and scheduling problems. Simulation is also used to test various dispatching rules for determining the real time flow of parts. Group technology ideas based on clustering analysis are used to form part-tool, machine-part groups or to form groups of other FMS components.

Since a comprehensive model design that incorporates all aspects of FMS planning and scheduling problems is not computationally tractable, subproblems are identified and solved separately but within an integrated structure. Hierarchical approaches and iterative solution procedures based on evaluation and use of feed-back information are suggested. Heuristic methods have been developed to obtain approximate solutions for the subproblems which may often turn out to be computationally burdensome by themselves. Especially, when more than one subproblem is dealt with in one formulation, development of efficient heuristics appears to be the best approach for obtaining "good" feasible solutions. Actually, finding a good feasible solution for a model which includes as many aspects of the problem as possible in an integrated manner

may be of more practical value than a model which solves a single subproblem to optimality. However, usually considerable computational effort is required to solve each problem separately and sequentially in an iterative manner.

Furthermore, models which are formulated to find optimal solutions involve aggregated values of the system parameters and do not include certain important aspects of the real problem they attempt to represent. This is inevitable due to the complex nature of the problems whose computational requirements are beyond the capability of today's computing technology to be practical. Hence the direction of research in FMS operating problems is towards the development of efficient heuristic procedures. Within the current state of the computational power of computers, there are still problems inherent to FMSs which did not receive enough research attention. Meeting due dates of the part-types processed is one such problem. As mentioned in Smith, Ramesh, Dudek, and Blair [1986], users of FMS technology in the U.S. indicate that meeting due dates is their primary concern. However, research efforts have been directed more to maximizing the production rate and most of the objective functions formulated are its surrogates. Maximizing production rate does not necessarily guarantee meeting due dates. In models where meeting due dates is considered, due date is usually incorporated to be one of the bi-criteria objectives. We believe that due date related criteria are essential in all FMS planning and scheduling decisions.

Lot sizing, that is determining the amount to be processed on the FMS for each selected part-type is an issue which has not received the attention it deserves. Usually, part-types are assumed to have predetermined lot sizes. Then the decision is whether to

select a part-type or not. However, lot size determination may be considered during part-type selection which will integrate the two subproblems and give more practical results. It may not be feasible to process a part-type at a predetermined lot size and as a result, that part-type may never be selected. But it may be feasible to process it at a smaller lot size which is determined when the part-type is selected. Same issue is also relevant in machine loading. Lot sizes of operations may be determined while they are being allocated to the different machines in the FMS.

Another problem whose importance has always been emphasized but which still requires more research attention is the integration of subproblems not only within the same level but also at different levels. Particularly the integration of planning and scheduling problems is very important. If an integrated structure cannot be constructed, the solutions for planning problems obtained at the expense of considerable computational efforts will not be of any practical help. Therefore the gap that exists between planning and scheduling problems deserves investigation. Efforts towards combining loading policies with certain dispatching rules for real time flow are definitely worth their expense. Design of simulation experiments to analyze the effects of loading decisions on system's performance should continue. Planning models should be designed with the objective that the output they provide be easily implemented in further scheduling decisions. As mentioned in Shanker and Tzen [1985 p. 580], "loading of FMS should pave the road for sequencing."

For integrating the FMS problems defined and tackled at different levels and also for dealing with their computational burden, artificial intelligence methods and techniques

can be used together with those of OR, all combined under the structure of an expert system or a decision support system as suggested in Solot [1990]. Although the design of such systems depend heavily on the particular FMS characteristics, a general framework can be constructed.

Hence, there are still problems in FMS environments that require further research. Research should continue for designing comprehensive models and developing solution procedures to tackle these problems. Although computing power restrictions limit their practical applicability, they prove invaluable in providing a better understanding of the system dynamics and inherent characteristics. With this understanding, it will be possible to exploit certain special model structures towards the development of efficient algorithms whose solutions provide a sound decision support to the managers of FMSs.

CHAPTER 3

FMS PLANNING MODELS

In this chapter we present five MIP models for the FMS planning problems. There are two main formulations each of which incorporates a different demand situation. They deal with the part-type selection and lot sizing problems in an integrated manner. They can be viewed as single-machine problems, since, the machine capacities are aggregated in one system capacity constraint. Similarly, the tool magazine capacity of the FMS is considered on an aggregate basis. There is no explicit consideration of the operations of the part-types. One generalization of this problem is the multi-machine problem in which each machine and therefore, the tool magazine of each machine, are treated separately. This generalized formulation incorporates the machine loading problem as well. Each part-type is now assigned to a particular machine and the required tools are loaded on that machine's tool magazine. The most general form of all is the multi-machine, multi-operation problem where each part-type requires a series of operations which can be assigned to different machines. In this case, each operation of a selected part-type is to be assigned to a machine at a lot size equal to the lot size of the associated part-type. Unlike the other cases, a part-type may visit more than one machine. The routing of a part-type within the FMS is determined according to the assignment of its operations to the machines. This multi-machine multi-operation formulation considers the FMS planning problem with most of its relevant features. In

the next section, we present these formulations starting with the basic models. In those models where part-type assignments are made, we assume that the total tool slot requirement of a part-type is not greater than the tool magazine capacity of a machine. In the models where operation assignments are made, the same assumption is valid for an operation. We also assume that there is only one tool magazine set up in each period. This assumption is justified in those FMS's where tool changing requires a considerable amount of time. In some FMS's "the entire system is shut down when tool changeovers are done" [Kim and Yano, 1989b, p.5].

3.1. Model 1: A Part-type Selection and Lot-sizing Model

3.1.1. Single Machine Case

The MIP model we present below considers a T-period planning horizon. In the beginning of the planning horizon, there are a number of part-types waiting to be processed in the FMS during the next T periods. The available FMS capacity may or may not be sufficient to process all these part-types at their demanded quantities. Hence a subset of the part-types need to be selected and the lot sizes at which they can be processed must be determined. All the part-types may not be of same importance. Some part-types may be more urgent than others due to the nature of the downstream or customer demand. Also, some part-types may have closer due-dates than others. Part-types that are relatively more important than the others need to be produced in the earlier periods of the planning horizon. Therefore, the selected part-types are to be scheduled with respect to their relative importance through the periods of the planning horizon.

These periods can be viewed as the set up periods because, only one tool magazine set up is allowed in each period. At the end of each period, tool magazines are reloaded according to the next period's requirements. There are two main resources considered in this model: the aggregate machine time capacity and the aggregate tool magazine capacity available in each period. These act as overall system restrictions on the part-type selection and lot sizing decisions.

In the following table we give the definitions of the parameters and variables which will be used in the MIP formulations.

Table 1. Model Variables and Parameters

Indices:		
Part-type	: $i = 1, \dots, N$	Tool : $l = 1, \dots, L$
Machine	: $j = 1, \dots, M$	Period : $t = 1, \dots, T$
Operation	: $k = 1, \dots, K$	
Parameters:		
b_i	: objective function weight of part-type i	
D_i	: total demand for part-type i over the entire planning horizon	
d_{it}	: demand of part-type i in period t	
s_l	: number of tool slots used by tool l	
p_i	: processing time per unit of part-type i	
p_{ik}	: processing time per unit of operation k of part-type i	
h_{it}	: inventory carrying cost for part-type i in period t	
b_{it}	: backorder cost of part-type i in period t	
M_t	: aggregate tool magazine capacity in period t	
M_{jt}	: tool magazine capacity on machine j in period t	
C_t	: aggregate machine capacity in period t	
C_{jt}	: capacity on machine j in period t	
$L(l)$: set of part_types which require tool l .	
Variables:		
X_{it}	: production lot size of part-type i in period t	
X_{ijt}	: production lot size of part-type i on machine j in period t	
X_{ikjt}	: production lot size of operation k of part-type i on machine j in period t	
y_{lt}	: 1 if tool l is loaded in period t , 0 otherwise	
y_{jlt}	: 1 if tool l is loaded on machine j in period t , 0 otherwise	
I_i^+	: inventory of part-type i in the end of period t	
I_{it}^-	: amount backordered of part-type i in period t	

Model 1 :

(M1)

$$\text{Min} \sum_t \sum_i -(T-t+1) b_i X_{it}$$

st

$$\sum_t X_{it} \leq D_i \quad \forall i \quad (1)$$

$$\sum_i p_i X_{it} \leq C_t \quad \forall t \quad (2)$$

$$\sum_{i \in L(t)} p_i X_{it} - C_t y_{lt} \leq 0 \quad \forall l, t \quad (3)$$

$$\sum_i s_i y_{lt} \leq M_t \quad \forall t \quad (4)$$

$$X_{it} \geq 0 \quad \forall i, t$$

$$y_{lt} \in \{0,1\} \quad \forall l, t$$

Constraint set (1) guarantees that the selected part-types are not produced more than their demands. Constraint set (2) represents the machine time limitations for each period. Since separate variables are used to define part-types and tools, the relationship between them is represented by constraint set (3). These coupling constraints consider the tool sharing among part-types and guarantee that all the required tools are loaded on the tool magazine before an operation is assigned to a machine. Formulation of the part-type tool relationships in this form is different from its classical formulation in the literature. Most of the time, it provides a tighter formulation of the problem. In addition, it provides a structure which can be exploited when designing efficient solution procedures. We show in subsequent sections that, due to this structure, the tool magazine and the coupling constraints become redundant in the LP relaxation of the problem. Constraint set (4) controls the tool magazine availabilities. Finally, there is a surrogate objective function which enables the model to distinguish between different

part-types with respect to their associated weights. Those part-types which are more urgent than others are given priorities so that these are processed in earlier periods. Furthermore, it is always preferable to process a part-type as early as possible during the planning horizon so that total makespan is minimized. However, the purpose of defining the objective function in this manner is not makespan minimization. As a matter of fact, most of the time, it may not be feasible to process all the part-types at their demands within the span of the planning horizon. In such cases the makespan will be equal to the length of the planning horizon which is T periods. In any case, the objective function serves the major purpose of prioritization among the part-types based on their relative importance.

In the above formulation, the aggregation of the machine capacities requires the implicit assumption of identical machines. This assumption is not unrealistic. Based on the results of a survey on 155 FMSs, Jaikumar and Wassenhove [1987, p.9] report that there is a "trend towards systems which have either one or two machine types, within a system."

In the literature part-type selection problem has been studied as a single period problem. Part-types are selected for immediate production and usually, tool magazine capacity is considered to be the only system limitation. In Hwang and Shogan [1989], a time capacity constraint is added to the problem. However, it assumes fixed lot sizes of the selected part-types. The above model formulates the part-type selection problem over a T -period planning horizon considering due-dates. The part-types are selected with respect to three criteria, relative importance (coming from the due-dates or order

specifications), tool slot requirements and processing time requirements. There are no assumed order quantities. The lot sizes of the part-types in each period are determined within the model. With these features, we believe that, the model is different from and more comprehensive than the existing part-type selection models in the literature.

In Chapter 4 we present a branch and bound algorithm, based on the LP relaxation of problem M1.

3.1.2. A Generalization of Model 1: Multi-machine Case

Problem M1, presented in the above section, deals with the part-type selection and lot sizing aspects of the FMS planning problem. Since the machine and magazine capacities are aggregated, assignment of the selected part-types to different machines in the FMS is not considered. In this section, we consider the machines in the FMS separately. In addition to part-type selection and determination of their lot sizes, we also determine the part-types to be processed on each machine and the tools to be loaded on each machine's tools magazine. In this multi-machine model, we also consider fixture allocations by treating fixtures as special machine types based on the following definition of "fixture time". We define the amount of time a part-type needs a fixture of a certain type, that is the fixture time of a part-type, as the processing time of that part-type plus the average travel time plus load/unload times. Since we are dealing with the part-types not with their operations, we implicitly assume that all the operations of a part-type are assigned to the same machine. In fact, FMSs where all operations of a part-type are assigned to the same machine are not uncommon. Stecke [1988] presents a real FMS where all part-types consists of a single operation which is performed in one mount and

in one machine sitting. In such environments the following model adequately represents part-type selection, lot sizing, and machine loading problems in an integrated form.

For the notation use in the following formulation refer to Table 1 in Section 3.1.

(GM1)

$$\begin{aligned} \text{Min} \quad & \sum_t \sum_i -(T-t+1) b_i \left(\sum_j X_{ijt} \right) \\ \text{s.t.} \quad & \sum_i \sum_j X_{ijt} \leq D_i \quad \forall i \quad (5) \end{aligned}$$

$$\sum_i p_i X_{ijt} \leq C_{jt} \quad \forall j, t \quad (6)$$

$$\sum_{i \in L(j)} p_i X_{ijt} - C_{jt} y_{ijt} \leq 0 \quad \forall i, j, t \quad (7)$$

$$\sum_i s_i y_{ijt} \leq M_{jt} \quad \forall j, t \quad (8)$$

$$X_{ijt} \geq 0 \quad \forall i, j, t$$

$$y_{ijt} \in \{0,1\} \quad \forall i, j, t$$

Constraint set (5) assures that the total size of the lots processed on different machines from the same part-type does not exceed its demand. Constraint set (6) guarantees that the available machine capacities are not exceeded for each period. Constraint set (7) are the coupling constraints which provide that the required tools are loaded on the machine's magazine when a part-type is assigned to that machine. They also handle tool sharing among part-types. The tool magazine restrictions for each machine in each period are represented by constraint set (8). The objective function, as in problem M1, prioritize the part-types with respect to their associated weights.

Model GM1 is a generalization of Model 1 (M1) where the machines in the FMS are treated separately. The solution procedure which we present in Chapter 4 for model M1 is extended to also solve model GM1.

3.2. Model 2: A Part-type Selection and Lot-sizing Model for Minimizing Inventory and Backorders

3.2.1. Single Machine Case

In Model 1 (M1) which we have presented in Section 3.1, the demand for a part-type has to be met in some period during the planning horizon, according to the relative importance of that part-type. Part-type demands are not specified per period but rather given for the entire planning horizon. In Model 2, the demand for the part-types in each period of the planning horizon is specified. Unlike Model 1, a part-type may be demanded at different periods and the due-date of a part-type is the end of the period when it is demanded. In this case, the objective is to meet the demand for all part-types in the periods they are demanded. However, the FMS capacity may not be sufficient to allow such a production schedule. Therefore, some part-types may have to be backordered. Some of these part-types may be produced at quantities less than their demands and some may not even be produced at all during the planning horizon. In some periods, it may be possible to produce a part-type more than its demand in that period and to carry inventory. This serves the purposes of increasing system utilization in periods of low demand and avoiding possible backorders in future periods of high demand. We define the objective function as the minimization of total inventory and backorder costs. Since the major goal is to meet the part-type demands, backordering is always more highly penalized than inventory carrying. Model 2 which we present below considers the part-type selection and lot sizing problems in such a demand situation. For the notation used in the following formulation refer to Table 1 in Section 3.1.

Model 2 :
(M2)

$$\text{Min } \sum_t \sum_i h_{it} I_{it}^+ + b_i I_{it}^-$$

st

$$I_{it-1}^+ - I_{it-1}^- + X_{it} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (9)$$

$$\sum_i p_i X_{it} \leq C_t \quad \forall t \quad (10)$$

$$\sum_{i \in L(t)} p_i X_{it} - C_t y_{it} \leq 0 \quad \forall l, t \quad (11)$$

$$\sum_l s_l y_{lt} \leq M_t \quad \forall t \quad (12)$$

$$X_{it} \geq 0 \quad I_{it}^+ \geq 0 \quad I_{it}^- \geq 0 \quad \forall i, t$$

$$y_{lt} \in \{0,1\} \quad \forall l, t$$

Constraint set (9) is classical inventory balance equations for the part-types in each period. Constraints (10), (11), (12) are similar to constraints (2), (3), and (4) in Model 1. The objective function minimizes the inventory and backorder costs to provide the production of each part-type in the period immediately preceding its scheduled due-date.

As it can be seen from its formulation, M2 is a more complex problem than problem M1. The branch and bound algorithm developed for problem M1 exploits a structure which is not inherent in problem M2. In Chapter 4, we present two branch and bound algorithms to solve problem M2.

3.2.2. A Generalization of Model 2: Multi-machine Case

In Section 3.1. we presented a generalization of Model 1 to a multi-machine case. As stated in that section, this generalization incorporates the machine loading problem into the model. In this section we present a similar generalization for Model 2.

For the notation used in the following formulation refer to Table 1 in Section 3.1.

(GM2)

$$\text{Min } \sum_t \sum_i h_{it} I_{it}^+ + b_i I_{it}^-$$

st

$$I_{it-1}^+ - I_{it-1}^- + \sum_j X_{ijt} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (13)$$

$$\sum_i p_i X_{ijt} \leq C_{jt} \quad \forall j, t \quad (14)$$

$$\sum_{i \in L(j)} p_i X_{ijt} - C_{jt} y_{ijt} \leq 0 \quad \forall l, j, t \quad (15)$$

$$\sum_i s_i y_{ijt} \leq M_{jt} \quad \forall j, t \quad (16)$$

$$X_{ijt} \geq 0 \quad I_{it}^+ \geq 0 \quad I_{it}^- \geq 0 \quad \forall i, j, t$$

$$y_{ijt} \in \{0,1\} \quad \forall l, j, t$$

The above model, GM2, combines the part-type selection, lot sizing and machine loading problems with an objective of minimizing total inventory and backorders. The objective function is the same as in model M2. Constraint set (13) is the classical inventory balance equations where the amount produced from a part-type in a period is given by the sum over all machines in the FMS. Constraints (14), (15), (16) are similar to their counterparts in model GM1. Since the problem structure of essentially stays the same, the solution procedure developed for Model 2 can be applied for solving model GM2 with slight modifications.

Both in models GM1 and GM2, the operations of the part-types are not considered explicitly. In fact, it is assumed that all operations of a selected part-type will be processed on the same machine. As we have discussed earlier, there are FMSs where the practice is to allocate machines to part-types. However, to consider operations of the

part-types in machine loading may be more effective in terms of increasing the utilization of both the machines and their tool magazines. In the next section we present a formulation which considers the operations of part-types in the machine loading problem.

3.3. An Inclusive Model for Part-type Selection, Lot-sizing, and Machine Loading

The model we present in this section deals with the same demand situation as in Model 2. The demand for the part-types in each period of the planning horizon is known and the aim is to try to satisfy demands just-in-time. Hence, the objective function and the demand constraint for the part-types in each period are the same as in Model 2 (M2). However, instead of assigning part-types, we now assign the operations of part-types to the machines in the FMS. We define an operation as a series of small operations which can be processed on the same machine without any fixture adjustments in between. Hence an operation may require more than one type of tool and different operations may require common tools. Once a part-type is selected, all its operations must be assigned to machines and the tools required to process these operations must be loaded on the machine's magazine. In the previous models, a part-type visits only one machine in the FMS. In this model, a part-type may visit several machines according to the route determined by the assignment of its operations to machines. The lot size of an operation is determined by the lot size of its associated part-type. All operations of a part-type are processed at an equal amount in order to avoid work-in-process (WIP) inventories between periods. It is possible to split operation lot sizes among machines. That is, an operation is allowed to be assigned to more than one machine to provide routing flexibility.

The MIP formulation developed to deal with these attributes of the machine loading problem requires the definition of new variables and new constraints. Although the resulting model is indeed a generalization of Model 2 (M2, GM2), it has a different and considerably more complex structure.

For the notation used in the following formulation refer to Table 1 in Section 3.1.

(IM)

$$\text{Min} \sum_t \sum_i h_{it} I_{it}^+ + b_i I_{it}^-$$

st

$$I_{it-1}^+ - I_{it-1}^- + X_{it} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (17)$$

$$\sum_j X_{ikjt} - X_{it} = 0 \quad \forall i, k, t \quad (18)$$

$$\sum_i \sum_k p_{ik} X_{ikjt} \leq C_{jt} \quad \forall j, t \quad (19)$$

$$\sum_{(i,k) \in L(l)} p_{ik} X_{ikjt} - C_{jt} y_{ljt} \leq 0 \quad \forall l, j, t \quad (20)$$

$$\sum_l s_l y_{ljt} \leq M_{jt} \quad \forall j, t \quad (21)$$

$$X_{it} \geq 0 \quad I_{it}^+ \geq 0 \quad I_{it}^- \geq 0 \quad \forall i, t$$

$$X_{ikjt} \geq 0 \quad \forall i, k, j, t$$

$$y_{ljt} \in \{0,1\} \quad \forall l, j, t$$

In the above model, different from the previous models, the part-types and operations of these part-types are defined using separate variables. This requires the definition of a new constraint which will link the part-types to their operations. The objective function and the demand constraint (17) are the same as in problem M2 and they are expressed in terms of part-type variables. Constraints (19) and (20) are similar to the machine capacity and coupling constraints in problem GM2; however, here they

are written in terms of the operation variables. Now, we are assigning operations to machines and the tools to be loaded on the magazines are determined according to these operation assignments. Constraint set (18) serves as the linking constraint between constraint (17) and constraints (19) and (20). It guarantees that there is no WIP from any part-type at the end of a period, that is all operations of a part-type are processed at a lot size equal to its production size. Constraint (21) is the tool magazine constraint.

The MIP formulation, IM, incorporates many important aspects of the FMS planning problem. Part-types are selected, their operations are assigned to machines together with the cutting tools they require. Tool sharing among operations is considered so that the same type of tool is not loaded on the same machine's tool magazine more than once. If the tool requirements of operations lead to a grouping of identically tooled machines, then those groups will be formed. Furthermore, the lot size of each selected part-type and the lot sizes of its operations on the machines they are assigned to are determined. Fixture allocations are made by treating fixtures as special machine types based on our definition of "fixture time". We define the amount of time an operation needs a fixture of a certain type, that is the fixture time of an operation as the processing time of that operation plus the average travel time plus load/unload times. A part-type may require more than one type of fixture for its different operations.

Such an integrated approach makes the size of the resulting model grow rather quickly for real-size problems and exact solution procedures may become computationally infeasible. Therefore, we developed a heuristic procedure to obtain good feasible solutions for problem IM. In the next section, we present the heuristic procedure.

3.4. A Heuristic Procedure

As stated earlier, the size of the MIP model presented in Section 3.3 grows very large rather quickly for real-size problems. This renders general purpose MIP procedures a computationally infeasible alternative for obtaining an optimal or near optimal solution. In this section we present a heuristic procedure for the FMS planning problem, IM. Since it is developed for the most general problem, IM, it is also applicable to problems M1, GM1 and M2, GM2. The solution obtained can also be used as an initial upper bounding solution in an exact enumeration scheme. As a matter of fact, we use this heuristic procedure in one of the branch and bound algorithms designed to solve problem M2. We will refer to this heuristic procedure as heuristic HP hereafter. A general outline of HP is given below.

Step 1 For period t , compute the just-in-time production size for each part-type considering any backorder from the previous period.

Sort these part-types in a nonincreasing order of one of the following priorities:

- a. Backorder cost
- b. Backorder cost / Processing time
- c. Backorder cost / Tool slot requirements

Select the first part-type in the list.

Step 2 If the list is exhausted, go to Step 7, otherwise continue.

Compute total tool slot requirements for this part-type.

If aggregate tool magazine capacity is not sufficient it is not possible to produce this part-type in this period. Select the next part-type in the list and go to

beginning of Step 2. Otherwise continue to Step 3.

- Step 3 Sort the operations of the selected part-type in a nonincreasing order of their processing times breaking ties arbitrarily.

Select the first operation in the list.

- Step 4 If the list is exhausted then all operations of the current part-type have been loaded, go to Step 6 to adjust operation lot sizes.

Otherwise compute the tool slot requirements of the selected operation on every machine which can process this operation, considering savings due to the tools that have already been loaded.

If there is no machine with sufficient tool magazine capacity, then it is not possible to produce this part-type in this period (since no WIP is allowed).

Select the next part-type in the part-types list and go to Step 2. Otherwise continue to Step 5.

- Step 5 Among all machines which have sufficient machine capacity assign the operation to the machine providing the maximum tool slot savings.

If there is no machine with sufficient capacity to process the operation in a single lot, split the lot among the machines starting with the one having the largest available capacity and continuing in a nonincreasing order of machine capacities.

If the maximum lot size that can be produced from this operation is less than its predetermined lot size, record its maximum lot size.

Update machine and tool magazine capacities accordingly. Select the next

operation in the list and go to Step 4.

- Step 6 Among all the operations of the current part-type, find the one with the minimum lot size and adjust the lot sizes of all other operations accordingly. (This is required to avoid WIP between periods.) Update the production size of this part-type and record any unproduced quantity.

Update the machine and tool magazine capacities and go to Step 2.

- Step 7 Sort all part-types which cannot be produced at the required production sizes in a nonincreasing order of one of the priorities given in step 1. Make a list of infeasible part-types.

Check previous periods starting from $t-1$ for available machine and tool magazine capacities.

If there is available capacity in any of these periods, assign the operations of part-types in the infeasible part-types list as described in steps 3 to 6.

Continue this step until either the unproduced amounts for all part-types are zero or there is no capacity left in any period.

- Step 8 For those part-types which cannot be produced enough to meet their demands in the end of period t , either by being produced in the current period or carried as inventory from the previous period, backorder the unproduced amount to the next period. Let $t = t + 1$, go to Step 1.

This heuristic approach will always find a feasible solution since backordering an item to the next planning period is allowed. Within the structure described above, it is possible to experiment with alternative rules for assigning operations to the machines.

In the literature various heuristic approaches which are implementations of bin packing rules (see Rajagopalan [1986], Kim and Yano [1987a]) are suggested for solving the machine loading problem . Various scheduling rules have also been studied (see Stecke and Talbot [1983]). Different rules were shown to perform better under different operating conditions. Therefore, in order to find the best rule in terms of the quality of the feasible solution it produces, alternative rules can be tested. As stated above, the current rule for selecting operations is Longest Processing Time (LPT). Machine selection rule differs with respect to system's congestion. If there is sufficient machine capacity to perform the operation at its predetermined lot size, then the machine that provides the maximum tool savings is selected. If the lot size has to be split among machines, then it is split among the machines having the largest available capacities in order to avoid too much splitting and tool duplication. In part-type selection, backorder cost of a part-type is essential since the main objective is meeting the due-dates. When selecting part-types for producing to inventory, the trade-off between the inventory carrying and backorder costs should be considered. Since we assume that backordering is always more expensive than carrying inventory, selection of part-types for producing to inventory is also made based on backorder costs.

To apply heuristic HP to models M2 and GM2, all that is needed is to set the number of operations to one and eliminate the steps that are no longer applicable. For problem M2 number of machines is also set to one. Similar adjustments are needed for problems M1 and GM1. Additionally, the steps required for determining inventory and backorder quantities need to be eliminated.

CHAPTER 4 SOLUTION PROCEDURES

4.1. Model 1

4.1.1. A Lower Bounding Scheme

Problem M1 presented in Section 3.1 can be viewed as a two dimensional bin packing problem when there is no tool sharing between part-types, where the processing times and the tool slots denote the dimensions [Kim and Yano, 1987b]. Since bin packing problems are known to be NP-complete [Garey and Johnson, 1979], some relaxation of problem M1, which will be easier to deal with, has to be considered in the development of an efficient solution procedure. For this purpose, we consider the following LP relaxation of problem M1, where the integrality restrictions on tool variables, y_h 's are removed.

(RM1)

$$\text{Min } \sum_i \sum_t -(T-t+1) b_i X_{it}$$

st

$$\sum_i X_{it} \leq D_i \quad \forall i \quad (22)$$

$$\sum_i p_i X_{it} \leq C_t \quad \forall t \quad (23)$$

$$\sum_{i \in L(t)} p_i X_{it} - C_t y_{ht} \leq 0 \quad \forall l, t \quad (24)$$

$$\sum_i s_i y_{ht} \leq M_t \quad \forall t \quad (25)$$

$$x_{it} \geq 0 \quad \forall i, t, \quad 0 \leq y_{ht} \leq 1 \quad \forall l, t$$

Lemma 1: In problem RM1, there exists an optimal solution in which

$$y_{it} = \frac{\left(\sum_{i \in L(t)} p_i X_{it} \right)}{C_t} \quad \forall i, t.$$

Proof: Let X_{it}^* for $i=1, \dots, N$, $t=1, \dots, T$ and y_{it}^* for $i=1, \dots, L$, $t=1, \dots, T$

be an optimal solution to problem RM1. Then $y_{it}^* \geq \frac{\sum_{i \in L(t)} p_i X_{it}^*}{C_t}$. If $y_{it}^* = \frac{\sum_{i \in L(t)} p_i X_{it}^*}{C_t}$ then

the proof is done. Hence assume that $y_{it}^* > \frac{\sum_{i \in L(t)} p_i X_{it}^*}{C_t}$ for at least one l and one

t . Now, let $y_{it} = \frac{\sum_{i \in L(t)} p_i X_{it}^*}{C_t}$ for all l, t . Then $y_{it}^* \geq y_{it}$ for all l, t . And, since y_{it}^* for all l, t , is optimal to problem RM1, it is feasible and satisfies constraints (25). Then y_{it} also satisfies constraints (25) for all l, t . Therefore y_{it} for all l, t , is feasible to problem RM1. It is also optimal since the objective value of problem RM1 is not a function of y_{it} for any l, t . ■

In RM1, setting $y_{it} = \frac{\sum_{i \in L(t)} p_i X_{it}}{C_t}$, for all l and t , and substituting in constraints (25),

we get the following new constraints (25a) which replaces constraints (25).

$$\sum_i (S_i p_i) X_{it} \leq M_t C_t, \quad \forall t \quad (25a)$$

where S_i is the total number of tool slots required by part-type i .

Lemma 2: Let $X = \{X_{it} \mid \sum_i p_i X_{it} \leq C_t, \forall t\}$ and

$\hat{X} = \{X_{it} \mid \sum_i (S_i p_i) X_{it} \leq M_t C_t, \forall t\}$. If $S_i \leq M_t$ for all i and t then $X \subseteq \hat{X}$, and constraints (25a) are redundant with respect to constraints (23).

Proof: Let $x \in X$, then $\sum_i p_i x_{it} \leq C_t$ for all t . Multiplying both sides of this inequality by M_t we get $\sum_i (M_t p_i) x_{it} \leq M_t C_t$ for all t . But for each i , $(S_i p_i) x_{it} \leq (M_t p_i) x_{it}$ for all t , since $S_i \leq M_t$ for all i and t . Then, $\sum_i (S_i p_i) x_{it} \leq M_t C_t$ holds for all t , implying that $x \in \hat{X}$. ■

Based on Lemmas 1 and 2, problem RM1 reduces to the following problem.

(SM1)

$$\begin{aligned} \text{Min } & \sum_t \sum_i -(T-t+1) b_i X_{it} \\ \text{st } & \sum_i X_{it} \leq D_i \quad \forall i \end{aligned} \quad (22)$$

$$\begin{aligned} \sum_i p_i X_{it} & \leq C_t \quad \forall t \quad (23) \\ X_{it} & \geq 0 \quad \forall i, t \end{aligned}$$

Now, consider the following subproblems $S(t)$ for $t = 1, \dots, T$.

$$\begin{aligned} \text{Min } & \sum_i -(T-t+1) b_i X_{it} \\ \text{st } & X_{it} \leq D_i^t \quad \forall i \quad (22t) \\ & \sum_i p_i X_{it} \leq C_t \quad (23t) \\ & X_{it} \geq 0 \quad \forall i, t \end{aligned}$$

where for each $i \in \{1, \dots, N\}$

$$\begin{aligned} D_i^1 &= D_i \quad \text{and} \\ D_i^t &= D_i - \sum_{l=1}^{t-1} X_{il}^* \quad \text{for } t = 2, \dots, T \end{aligned}$$

where $X_i^* = (X_{i1}^*, \dots, X_{iT}^*)$ is an optimal solution for problem $S(t)$, for $t = 1, \dots, T$. In order to solve problem SM1 we have devised an algorithm which we will refer as algorithm ALGM hereafter. This algorithm constructs an optimal solution,

$X^* = (X_1^*, \dots, X_T^*)$, to problem SM1, by sequentially solving problems $S(t)$, $t = 1, \dots, T$, starting with $S(1)$. The validity of algorithm ALGM is given in Theorem 1.

Theorem 1: For $t = 1, \dots, T$, let $X_t^* = (X_{1t}^*, \dots, X_{Nt}^*)$ be an optimal solution for problem $S(t)$, where these solutions are obtained sequentially starting with $t=1$ and ending with $t=T$. Also let $X^* = (X_1^*, \dots, X_T^*)$, then X^* is an optimal solution for problem SM1.

In order to prove Theorem 1, we will define a new problem. Based on this new problem, we will state and prove two lemmas which in turn will be used in the proof of Theorem 1.

Let $c_u = (T-t+1)b_t$ and consider the following relaxation, LM, of problem SM1 in which constraints (23) are dualized with multipliers $u_i \geq 0$ for $i = 1, \dots, N$. For notational convenience, we write the objective function as a maximization problem. (LM)

$$\begin{aligned} & \text{Max } \sum_t \sum_i (c_u - u_i) X_{it} + u_i D_i \\ & \text{st } \sum_i p_i X_{it} \leq C_t \quad \forall t \\ & \quad X_{it} \geq 0 \quad \forall i, t \end{aligned} \quad (24)$$

From LP duality, for given $u = (u_1, \dots, u_N) \geq 0$, any optimal solution to problem LM is optimal to problem SM1, if this solution is feasible in SM1 and satisfies the complementary slackness. Clearly, X^* is feasible in problem SM1 and in problem LM. Therefore in order to prove that X^* is optimal in SM1, we need to show that there exist a vector $u = (u_1, \dots, u_N) \geq 0$ such that :

1. X^* is optimal in LM and

$$2. \quad u_i \left(D_i - \sum_i X_i^* \right) = 0 \text{ for all } i.$$

Preliminaries: Since subproblems $S(t)$ ($t=1, \dots, T$) are continuous upper bounded knapsack problems, their optimal solutions are determined according to the c_i/p_i ratios. Furthermore, it is a well-known fact that $S(t)$ ($t=1, \dots, T$) possesses optimal solutions where a part is produced either at its upper bound or as much as constraints (22t) permit. $S(t)$'s ($t=1, \dots, T$) are solved sequentially and parts which are produced at their upper bounds are eliminated from subsequent subproblems. Note that parts produced in period t have higher c_i/p_i ratios than parts produced in period $t+j$ ($j=1, 2, \dots, T-t$), for any t . It must be clear that, for any t , determining the priorities of parts with respect to their c_i/p_i ratios is the same as determining their priorities with respect to their b_i/p_i ratios. For notational convenience we will use b_i/p_i when we refer to the priority of part i for any fixed t .

If we substitute the optimal solution X^* which is obtained by solving subproblems $S(t)$ sequentially for $t \in \{1, 2, \dots, T\}$ in problem SM1, two possible cases may arise:

- (i) All demand constraints (22) are binding and for at least one t , constraint (23) (equivalently constraint (24) in problem LM) has a positive slack
- (ii) Constraints (23) are binding but there may or may not be positive slacks in the demand constraints.

We first look at Case (i).

Case (i): Let $I(t) = \{i \mid X_i^* > 0\}$ for all t . Assume that z is the first period where constraint (23) has a positive slack. This implies that constraints (23) for $t = 1, \dots, z-1$ are binding. Furthermore constraints (23) for $t = z+1, \dots, T$ have slacks equal to their right hand side values, that is $I(t) = \emptyset$ for $t = z+1, \dots, T$. This follows from the bounded knapsack problem structure of problems $S(t)$, $t \in \{1, \dots, T\}$, and the sequential manner with which we solve them. Note that the cardinality of the set $\{I(t) \cap I(t+1)\}$ for $t \in \{1, \dots, T-1\}$ is at most one.

We now give a formal procedure for obtaining the u_i^* , $i \in \{1, \dots, N\}$ values.

Step 0 Initialization. Set $r_z = \frac{c_{iz} - u_i}{p_i} = 0$ for all $i \in I(z)$, and solve for u_i^* to obtain $u_i^* = c_{iz}$. Set $t = z$.

Step 1 Set $t = t - 1$.

Determine $m = \underset{i \in I(t)}{\operatorname{argmin}} \left(\frac{b_i}{p_i} \right)$.

Choose u_m^* to satisfy $\frac{c_{mt+1} - u_m^*}{p_m} = r_{t+1}$.

Step 2 Set $r_t = \frac{c_{mt} - u_m^*}{p_m}$.

Solve $\frac{c_{it} - u_i}{p_i} = r_t$ for u_i^* for all $i \in I(t)$.

Step 3 If $t = 1$, Stop. All u_i^* , $i \in \{1, \dots, N\}$ have been determined.

Otherwise go to step 1.

Note that in step 2, $u_i^* = c_{it} - \frac{p_i}{p_m} (c_{mt} - u_m^*) \quad \forall i \in I(t)$.

Lemma 3: When Case (i) holds and the multipliers, u^* , are determined as above, then X^* is an optimal solution to problem LM.

Proof: Select an arbitrary period $t \in \{1, 2, \dots, T\}$. For period t , we have to show that $\frac{c_{it} - u_i^*}{p_i} \leq r_t$ for all $i \notin I(t)$. That is, we have to show:

$$(a) \quad \frac{c_{it} - u_i^*}{p_i} \leq r_t \quad \forall i \in I(z), \quad \forall z=1, \dots, t-1$$

$$(b) \quad \frac{c_{it} - u_i^*}{p_i} \leq r_t \quad \forall i \in I(z), \quad \forall z=t+1, \dots, T$$

We will show both (a) and (b) by induction. To show (a) by induction, we will first show that (a) is true for $z = t-1$.

Let $m = \underset{i \in I(t-1)}{\operatorname{argmin}} \left(\frac{b_i}{p_i} \right)$ then, by the construction of u^* ,

$$\frac{c_{mt} - u_m^*}{p_m} = r_t \quad \text{and} \quad r_{t-1} = \frac{c_{mt-1} - u_m^*}{p_m}$$

Hence for all $i \in I(t-1)$,

$$\frac{c_{it-1} - u_i^*}{p_i} = r_{t-1} = \frac{c_{mt-1} - u_m^*}{p_m} \quad \text{and} \quad u_i^* = c_{it-1} - \frac{p_i}{p_m} (c_{mt-1} - u_m^*).$$

Is $\frac{c_{it} - u_i^*}{p_i} \leq r_t = \frac{c_{mt} - u_m^*}{p_m}$ for all $i \in I(t-1)$?

Substituting in u_i^* we get

$$\frac{c_{it} - c_{it-1}}{p_i} + \frac{c_{mt-1} - c_{mt}}{p_m} \leq 0 \quad \text{for all } i \in I(t),$$

where $c_{it} - c_{it-1} = -b_i$ and $c_{mt-1} - c_{mt} = b_m$. Hence we have

$$-\frac{b_i}{p_i} + \frac{b_m}{p_m} \leq 0 \quad \text{or} \quad \frac{b_i}{p_i} \geq \frac{b_m}{p_m} \quad \text{for all } i \in I(t-1).$$

The last inequality is true since $m = \underset{i \in I(t-1)}{\operatorname{argmin}} \left[\frac{b_i}{p_i} \right]$. Hence (a) is true for $z = t-1$. Now assume that (a) follows for $z = t-n$ for any $n = 1, \dots, t-2$. We will show that it follows for $z = t-n-1$.

Let $k = \underset{i \in I(t-n-1)}{\operatorname{argmin}} \left[\frac{b_i}{p_i} \right]$ then, by the construction of u^* ,

$$\frac{c_{k(t-n)} - u_k^*}{p_k} = r_{t-n} = \frac{c_{i(t-n)} - u_i^*}{p_i} \text{ for any } i \in I(t-n)$$

$$u_k^* = \frac{c_{k(t-n)} - p_k}{p_i} (c_{i(t-n)} - u_i^*) \text{ for any } i \in I(t-n)$$

and

$$\frac{c_{j(t-n-1)} - u_j^*}{p_j} = r_{t-n-1} = \frac{c_{k(t-n-1)} - u_k^*}{p_k} \quad \forall j \in I(t-n-1)$$

$$u_j^* = c_{j(t-n-1)} - \frac{p_j}{p_k} (c_{k(t-n-1)} - u_k^*) \quad \forall j \in I(t-n-1)$$

$$\text{Is } \frac{c_{jt} - u_j^*}{p_j} \leq r_t = \frac{c_{mt} - u_m^*}{p_m} \text{ for all } j \in I(t-n-1) \text{ and } m \in I(t) ? \quad (1a)$$

Substituting u_k^* in u_j^* and u_j^* in (1a) we get

$$\frac{c_{jt} - c_{j(t-n-1)}}{p_j} + \frac{c_{k(t-n-1)} - c_{k(t-n)}}{p_k} + \frac{c_{i(t-n)} - u_i^*}{p_i} - \frac{c_{mt} - u_m^*}{p_m} \leq 0 \quad (2a)$$

where $i \in I(t-n)$, $k, j \in I(t-n-1)$, $m \in I(t)$, and

$$c_{jt} - c_{j(t-n-1)} = -(n+1)b_j$$

$$c_{k(t-n-1)} - c_{k(t-n)} = b_k$$

$$c_{i(t-n)} - c_{it} = nb_i$$

Substituting these in (2a) we get

$$-\frac{(n+1)b_j}{p_j} + \frac{b_k}{p_k} + \frac{nb_i}{p_i} + \frac{c_{it} - u_i^*}{p_i} - \frac{c_{mt} - u_m^*}{p_m} \leq 0 \quad (3a).$$

The last two terms in (3a) will give a nonpositive value by induction hypothesis:

$$\frac{c_{it} - u_i^*}{p_i} \leq \frac{c_{mt} - u_m^*}{p_m} \text{ for } i \in I(t-n).$$

We also know that

$$\frac{b_j}{p_j} \geq \frac{b_k}{p_k} \text{ since } k = \underset{i \in I(t-n-1)}{\operatorname{argmin}} \left\{ \frac{b_i}{p_i} \right\} \text{ and } \frac{b_j}{p_j} \geq \frac{b_i}{p_i} \text{ since } i \in I(t-n) \text{ and } j \in I(t-n-1).$$

Therefore, $\frac{(n+1)b_j}{p_j} \geq \frac{b_k}{p_k} + \frac{nb_i}{p_i}$ and the first three terms will also give a nonpositive value. Hence (a) follows for $(t-n-1)$ which completes the proof for (a). ■

To show (b) by induction, first we will show that (b) holds for $z = t+1$.

Let $m = \underset{i \in I(t)}{\operatorname{argmin}} \left\{ \frac{b_i}{p_i} \right\}$ then by the construction of u^* ,

$$\frac{c_{i(t+1)} - u_i^*}{p_i} = r_{t+1} = \frac{c_{m(t+1)} - u_m^*}{p_m} \text{ and } u_i^* = c_{i(t+1)} - \frac{p_i}{p_m} (c_{m(t+1)} - u_m^*) \quad \forall i \in I(t+1)$$

Is $\frac{c_{it} - u_i^*}{p_i} \leq r_t = \frac{c_{mt} - u_m^*}{p_m}$ for all $i \in I(t+1)$?

Substituting in u_i^* we get

$$\frac{c_{it} - c_{i(t+1)}}{p_i} - \frac{c_{mt} - c_{m(t+1)}}{p_m} \leq 0 \text{ for all } i \in I(t+1).$$

where $c_{it} - c_{i(t+1)} = b_i$, $c_{mt} - c_{m(t+1)} = b_m$. Hence we have $\frac{b_i}{p_i} - \frac{b_m}{p_m} \leq 0$ or

$$\frac{b_i}{p_i} \leq \frac{b_m}{p_m}. \text{ This last inequality is true since } i \in I(t+1) \text{ and } m \in I(t). \text{ Hence (b)}$$

follows for $z = t+1$.

Now assume that (b) follows for $z = t+n$ for any $n = 1, \dots, T-t-1$. We will show that it follows for $z = t+n+1$.

Let $k = \underset{i \in I(t+n)}{\operatorname{argmin}} \left(\frac{b_i}{p_i} \right)$, then by the construction of u^* ,

$$\frac{c_{i(t+n+1)} - u_i^*}{p_i} = r_{t+n+1} = \frac{c_{k(t+n+1)} - u_k^*}{p_k} \quad \forall i \in I(t+n+1)$$

$$u_i^* = c_{i(t+n+1)} - \frac{p_i}{p_k} (c_{k(t+n+1)} - u_k^*) \quad \forall i \in I(t+n+1)$$

$$\text{Is } \frac{c_{ii} - u_i^*}{p_i} \leq r_t = \frac{c_{mi} - u_m^*}{p_m} \text{ for all } i \in I(t+n+1) \text{ and } m \in I(t) ? \quad (1b)$$

Substituting in u_i^* in (1b) we get

$$\frac{c_{ii} - c_{i(t+n+1)}}{p_i} + \frac{c_{k(t+n+1)} - u_k^*}{p_k} - \frac{c_{mi} - u_m^*}{p_m} \leq 0 \quad (2b)$$

where $i \in I(t+n+1)$, $m \in I(t)$, and $k \in I(t+n)$ and

$$\begin{aligned} c_{ii} - c_{i(t+n+1)} &= (n+1)b_i \\ c_{k(t+n+1)} - c_{ki} &= -(n+1)b_k \end{aligned}$$

Substituting these in (2b) we get

$$\frac{(n+1)b_i}{p_i} - \frac{(n+1)b_k}{p_k} + \frac{c_{ki} - u_k^*}{p_k} - \frac{c_{mi} - u_m^*}{p_m} \leq 0 \quad (3b)$$

The last two terms in (3b) will give a nonpositive value by induction hypothesis:

$$\frac{c_{ki} - u_k^*}{p_k} \leq \frac{c_{mi} - u_m^*}{p_m} \text{ for all } i \in I(t+n), m \in I(t).$$

We know that $\frac{b_i}{p_i} \leq \frac{b_k}{p_k}$ since $i \in I(t+n+1)$, $k \in I(t+n)$. Therefore, the first two terms will also give a nonpositive value. Hence (b) follows for $z = t+n+1$ which completes the proof for (b). ■

By (a) and (b) we have shown that, for period $t \in \{1, \dots, T\}$, $\frac{c_{it} - u_i^*}{p_i} \leq r_t$ for

all $i \notin I(t)$. Since t is an arbitrary period this inequality holds for all $t \in \{1, \dots, T\}$. Furthermore, recall that for any fixed $t \in \{1, \dots, T\}$, u_i^* 's are chosen to satisfy

$\frac{c_{it} - u_i^*}{p_i} = r_t$ for all $i \in I(t)$ in problem LM. Therefore, for any fixed $t \in \{1, \dots, z-1\}$,

any set of positive values for X_{it} for all $i \in I(t)$, that makes constraints (26) binding are optimal in LM. For $t = z$ where there is a positive slack in constraint (26), $r_z = 0$, hence any set of positive values for X_{iz} for all $i \in I(z)$, that satisfy constraint (26) are optimal in problem LM. Since X^* is one such solution, then X^* is optimal for problem LM. This completes the proof for Case (i). ■

Case (ii): Let $S = \{i \mid \sum_{t=1}^T X_{it} < D_i\}$. Set $u_i^* = 0$ for all $i \in S$. Let

$r_T = \frac{c_{mT} - u_m^*}{p_m} = \frac{c_{mT}}{p_m}$ where $m \in I(T) \cap S$. Then, for all $i \in I(T)$ and $i \neq m$, set

$$u_i^* = c_{iT} - \frac{p_i}{p_m} c_{mT}.$$

If $I(T) \cap S = \emptyset$ then let $r_T = 0$ and set $u_i^* = c_{iT}$ for all $i \in I(T)$.

Now, the u_i^* values for all $i \in I(t)$ for $t=1, \dots, T-1$, can be computed as in Case (i) by starting with step 1, and setting $t = T-1$.

Lemma 4: When Case (ii) holds and the multiplier values are determined as above, then X^* is an optimal solution to problem LM.

Proof: Proof of Lemma 4, is similar to that of Lemma 3 and is therefore omitted.

Proof of Theorem 1: By Lemmas 3 and 4, we have shown that X^* is an optimal solution to problem LM with multipliers u^* . Since these multipliers are determined so

as to satisfy the complementary slackness in problem SM1 and since X^* is feasible in SM1, then X^* is an optimal solution for problem SM1. ■

By the proof of Theorem 1, we have shown that, problem SM1 can be decomposed into T subproblems, $S(t)$ ($t=1, \dots, T$), which when solved sequentially starting with $t=1$ and ending with $t=T$, provides an optimal solution to problem SM1. An attractive property of these subproblems is that they are continuous knapsack problems with bounded variables and can be solved very efficiently. In implementing the algorithm, ALGM, subproblems $S(t)$, $t = 1, \dots, T$, are solved with the Weighted Selection Procedure (WSP) [Faaland, 1984]. We now present a formal statement of algorithm ALGM.

- Step 1. Set $t = 1$.
- Step 2. Solve $S(t)$ with WSP to find the optimal solution X_t^* .
- Step 3. Update D_i' for $i = 1, \dots, N$. If $D_i' = 0$ for all i , then set $X_v^* = 0$ for $v = t+1, t+2, \dots, T$ and go to step 5.
- Step 4. Set $t = t + 1$ and go to step 2.
- Step 5. Terminate. $X^* = (X_1^*, \dots, X_T^*)$ is an optimal solution for problem M.

The following remarks provide further insight into ALGM and help establish its worst case computational bound.

Remark 1 [Faaland, 1984]: A continuous knapsack problem with no more than N bounded variables is solvable in $O(N)$ time by the WSP.

Remark 2: If the demand for part i , $i \in \{1, \dots, N\}$, is completely satisfied when $S(t)$ is solved, then the variables X_u for $t+1, \dots, T$ are eliminated from the

subsequent knapsack problems. It follows from this observation that if at some period $z < T$, the demand for all of the N parts have been satisfied, then ALGM terminates after solving z knapsack problems. Consequently, ALGM terminates after solving at most T continuous knapsack problems with bounded variables.

Theorem 2: The computational complexity of ALGM is $O(NT)$.

Proof: The proof easily follows from Remarks 1 and 2 and the fact that the demand update step of ALGM (step 3) is done in $O(N)$ time.

Algorithm ALGM provides a very efficient way of solving problems SM1 which is the LP relaxation of problem M1. In order to find an exact solution to problem M1, we have developed a branch and bound algorithm. Hereafter we will refer to this algorithm as algorithm ALG1. In ALG1, the lower bounding subproblems solved at the nodes of the branch and bound tree are in the form of problem SM1 and therefore solved by using algorithm ALGM. In the next section we will give an informal description of algorithm ALG1 followed by its formal statement.

4.1.2. Algorithm ALG1

Problem M1 presented in Section 3.1 is a MIP problem, where the number of integer variables easily grow large for problems of real world dimensions. Therefore, it may not be possible to find even a feasible solution to the problem by using standard MIP packages. The branch and bound algorithm we will now present solves the problem optimally in a very efficient manner.

4.1.2.1. An informal description of algorithm ALG1

At each node of the branch and bound tree, we solve the LP relaxation, RM1, of problem M1, which reduces to problem SM1 as it is shown in Section 4.1.1. We have also shown that algorithm ALGM solves this problem in $O(NT)$ time. At each node of the branch and bound tree we need to find optimal solutions to T upper bounded continuous knapsack problems. However, due to the branching strategy we employ, it is not required to solve all T problems at each node. At the root node, we solve all T subproblems with all the variables free and obtain the optimal solution values, X_{it}^* . If this solution satisfies the tool magazine constraint we stop. Otherwise, we branch on the X_{it} variables starting with $t=1$ and i having the largest b_i/p_i ratio among those which have positive optimal solution values. On the right branch we set $X_{it} = 0$. On the left branch we set $X_{it} > 0$. However, since X_{it} is already positive at the parent node, i.e. $X_{it} = X_{it}^*$, then the solution obtained at the parent node is feasible to the problem with the additional constraint $X_{it} > 0$, therefore it is also optimal. Consequently, we do not need to solve the subproblem at the left node. We just set $X_{it} = X_{it}^*$ and proceed through the left node. We set all the tool variables of those tools required by part-type i to 1 and check the tool magazine feasibility. If the available tool magazine capacity is sufficient to hold the required tools, we update the tool magazine capacities. While checking for the tool magazine feasibility, we consider the tools which are already loaded on the magazine. Then we proceed branching on the part-type i having the next largest b_i/p_i and a positive optimal solution value. We continue through the left nodes in this manner until the solution becomes infeasible with respect to the tool magazine capacity. Then

we backtrack. While we are moving down the left nodes, if all the part-types with positive X_{it}^* values have been fixed in a period, we continue branching on the X_{it} variables in the next period. If all the periods have been exhausted we backtrack. This branching strategy enables us to solve the subproblems only at the root node and at the nodes we backtrack to. Hence, algorithm ALG1 solves the lower bounding subproblems only at half of the total number nodes it creates.

In the branching strategy employed, the sequence in which the variables are branched upon is very essential. As described in the above paragraph, ALG1 starts branching with the first time period and with the part-type having the largest b_i/p_i ratio and continues in a nonincreasing order of these ratios and a nondecreasing order of periods. Due to this branching sequence, when solving a lower bounding subproblem at a node, we need to solve a continuous knapsack problem for each period, starting with the period corresponding to this node and ending with period T . When we backtrack to a node, we check the period t' corresponding to this node. We then solve subproblems $S(t)$ for $t = t', \dots, T$ since subproblems $S(t)$ for $t = 1, \dots, t'-1$ have already been solved before this node is reached. Furthermore, for period t' , the variables that have been fixed at some positive values along the path from the initial node to the current node, are eliminated from the lower bounding subproblem $S(t')$. This branching strategy is based on the separability of the lower bounding subproblem SM1, with respect to time periods and also on its knapsack structure.

At any node, if the lower bound obtained is greater than the current upper bound we fathom and backtrack. Otherwise, we check the tool magazine feasibility of this

lower bounding solution. If the solution is feasible an upper bound is obtained. In this case, we fathom that node and backtrack again. Otherwise we proceed in the same manner as we did down the root node. Note that, we branch down a node only if the lower bounding solution is infeasible to the original problem. Therefore, while branching down the left nodes upon those variables which have positive values in the lower bounding solution, we know that it will eventually become infeasible to fix one more variable at some positive value. The algorithm ends when all the nodes are fathomed.

4.1.2.2. A formal statement algorithm ALG1

- Step 0 Set $n = 0$ (number of the stored node) and $t' = 1$. Set an upper bound.
- Step 1 Solve the subproblems $S(t)$ for all $t = t', \dots, T$. Compute the lower bound. If the lower bound is greater than the upper bound go to Step 6. If the solution is feasible update the upper bound and the incumbent solution if necessary and go to Step 6. Otherwise continue.
- Step 2 For each period $t = t', \dots, T$, find those part-types that have positive optimal values, in the solution obtained in Step 1. Sort these part-types in a nonincreasing order of their b_i/p_i ratios to obtain a list for each period $t = t', \dots, T$. Set $t = t'$.
- Step 3 a. Select the first part-type in the list for period t . If the list for period t is exhausted set $t = t + 1$. Go to Step 3b.
- Otherwise eliminate the selected part-type from the list for period t . Go to Step 4.
- b. If $t = T + 1$, go to Step 6. Otherwise go to Step 3a.

- Step 4 Branch on the selected part-type i . At the left branch set $X_{it} = X_{it}^*$ and at the right branch set $X_{it} = 0$. Let $n = n + 1$. Store the required information at node n on the right branch. Continue to Step 5.
- Step 5 On the left branch, set the tool variables of those tools required by the selected part-type to one. Check the tool magazine capacity considering the tools already loaded. If there is sufficient capacity update the tool magazine capacity. To select the next part-type in the list, go to Step 3a. Otherwise go to Step 6.
- Step 6 Fathom that node. Backtrack to node n . If $n = 0$ STOP, current solution is optimal. Otherwise, check the period t at node n .
Set $t' = t$. Go to Step 1.

It must be clear that the algorithm presented above will yield an exact optimal solution. This follows from the validity of our lower and upper bounds and the fact that no solution is ever excluded from consideration. Note that we implicitly enumerate all the solutions by considering the two possibilities for each part-type in each period:

- a) The part-type is produced at some positive level.
- b) The part-type is not produced.

Lemmas 1 and 2 directly extend to problem GM1. Based on these, algorithm ALG1 described above for problem M1, can also be used for problem GM1 with slight modifications. The only difference is in the consideration of different machines to which part-types are assigned. The lower bounding subproblems have the same structure in both problems and the same branching and fathoming rules are valid. The number of

variables to branch on increase by a factor of M , which is the number of machines. The computational results obtained by solving randomly generated problems both for problem M1 and problem GM1 are given in Chapter 5.

4.2. Model 2

4.2.1. A Lower Bounding Scheme

Model 2 (M2) presented in Section 3.2 has a more complex structure than Model 1 (M1) due to the form of the objective function and the demand constraints. In the solution procedure developed for problem M1, solutions to the lower bounding subproblems are obtained by solving T upper bounded knapsack problems. This is induced by the structure of the objective function and demand constraints. In problem M2, this structure is no longer preserved. Therefore, it is required to design new solution procedures to be able to efficiently solve to the lower bounding subproblems. These lower bounding subproblems are based on the LP relaxation of problem M2 as in the branch and bound algorithm developed for problem M1. By Lemma 1 and Lemma 2, given in Section 4.1.1, the LP relaxation of problem M2 reduces to the following problem.

(SM2)

$$\begin{aligned} & \text{Min } \sum_t \sum_i \left(h_{it} I_{it}^+ + b_{it} I_{it}^- \right) \\ & \text{s.t.} \\ & I_{it-1}^+ - I_{it-1}^- + X_{it} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (25) \end{aligned}$$

$$\sum_i p_{it} X_{it} \leq C_t \quad \forall t \quad (26)$$

$$X_{it} \geq 0, \quad I_{it}^+ \geq 0, \quad I_{it}^- \geq 0 \quad \forall i, t$$

Problem SM2 is a linear programming problem and therefore it can be solved directly by using a linear programming algorithm. Later in this chapter we present a branch and bound algorithm, ALG2, in which the subproblems SM2 are solved in this manner. However, SM2 also has the structure of a network problem with side constraints. It is possible to consider the following relaxation of problem SM2 in which constraint (26) is relaxed using multipliers, v .

(RSM2)

$$\begin{aligned} \text{Min } & \sum_i \sum_t \left(h_{it} I_{it}^+ + b_{it} I_{it}^- \right) + \sum_i v_i \left(\sum_t p_{it} X_{it} \right) - \sum_i v_i C_i \\ \text{st } & I_{it-1}^+ - I_{it-1}^- + X_{it} - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (27) \\ & X_{it} \geq 0, \quad I_{it}^+ \geq 0, \quad I_{it}^- \geq 0 \quad \forall i, t \end{aligned}$$

The above problem has a network structure which, for fixed multipliers values, can be solved by using the algorithm presented in Erenguc and Tufekci [1988]. We will refer to this algorithm as algorithm NET hereafter. Some slight modifications are required to implement the algorithm however, its computational complexity, which is $O(NT^2)$, does not change. Hence, we use this algorithm for solving Lagrangian subproblems RSM2 for fixed multiplier values and apply subgradient optimization for multiplier adjustment in order to maximize the Lagrangian dual. In this manner we obtain a lower bound to problem SM2. This lower bounding scheme is then imbedded in a branch and bound algorithm, ALG3, which is similar to ALG2. In the next section we present an informal description of the branch and bound algorithms ALG2 and ALG3, followed by their formal statements.

4.2.2. Algorithms ALG2 and ALG3

The branch and bound algorithms ALG2 and ALG3 are both developed to find an exact solution to problem M2. The general structure of the two algorithms is similar to algorithm ALG1 presented in Section 4.1.2. However, the lower bounding subproblems have considerably different structures. Since problem SM2 is not separable with respect to time periods and does not have a knapsack problem structure it is no longer possible to use algorithm ALGM for its solution. This also leads to differences in the branching strategies.

4.2.2.1. An informal description of algorithm ALG2

In algorithm ALG2 we use a linear programming algorithm to solve the lower bounding subproblems SM2. At the initial node we solve SM2 with all variables free and obtain an optimal solution X_{it}^* . We then check the tool magazine feasibilities. If this solution is feasible to the original problem we stop. Otherwise we branch on the X_{it}^* variables whose values are positive. On the right branch, we set $X_{it} = 0$. On the left branch we set $X_{it} > 0$. As in algorithm ALG1, we do not need to solve any subproblems at the left nodes, since we always branch on a variable which has a positive solution value at the parent node and this solution carries down to child nodes. Therefore, on the left nodes we only check tool magazine feasibility considering the tools that have already been loaded. If it is feasible to set the variable X_{it} to some positive value with respect to the tool magazine capacity, we continue branching down the left node. As we branch down the left nodes we store information on the right nodes to be considered later. We backtrack when no more X_{it} variables can be set to positive values

due to tool magazine infeasibility. At each node we backtrack to we solve the lower bounding subproblem SM2 and check the feasibility of the solution in terms of tool magazine capacity. If this solution is feasible to the original problem we fathom that node and backtrack. Therefore, we branch down a node only when the solution to the subproblem solved at that node is infeasible to the original problem. Consequently, when we keep branching down the left nodes, we know that it will eventually become infeasible to set a X_{it} variable to a positive value. This structure of algorithm ALG2 is very similar to algorithm ALG1. However, unlike algorithm ALG1, we cannot set $X_{it} = X_{it}^*$ at the left nodes, since the subproblem SM2 is not separable with respect to time periods and does not have a knapsack problem structure. Therefore, when the algorithm backtracks to a node, it is no longer possible to eliminate the variables that have been fixed to some positive values from the lower bounding subproblem SM2. Instead, when solving the subproblem SM2 at any node, variables which are restricted to be positive at that node are relaxed to accept nonnegative values. This is a valid relaxation and it therefore provides us a lower bound. In algorithm ALG1, the sequence in which the variables are branched on is very important. This sequence is possible by the knapsack problem structure of SM1 and its separability with respect to time periods. These properties of SM1 make it possible to fix the X_{it} variables at some positive values and eliminate them from further consideration. However in algorithm ALG2, this is no longer possible and hence the sequence in which the variables are branched has no particular significance. For computational convenience we start branching with period $t=1$ and part-type $i=1$ continue in an increasing order of these indices. Other fathoming

criteria are same as in algorithm ALG1, and the algorithm terminates when all nodes are fathomed.

4.2.2.2. A formal statement of algorithm ALG2

- Step 0 Set $n = 0$ (number of the stored node). Set an upper bound.
- Step 1 Solve the lower bounding subproblem SM2 using a linear programming algorithm. If the lower bound is greater than the upper bound go to Step 5. If the solution is feasible update the upper bound and the incumbent solution if necessary and go to Step 5. Otherwise set $t' = 1$ and continue to Step 2.
- Step 2 a. Select a variable $X_{it'}$ and which has a positive value in the optimal solution obtained in Step 1. If all $X_{it'}$ variables with positive solution values have been branched on go to Step 2b.
- b. Set $t' = t' + 1$. If $t' = T + 1$ go to Step 5. Otherwise go to Step 2a.
- Step 3 Branch on the selected variable. At the left branch set $X_{it} = X_{it}^*$ and at the right branch set $X_{it} = 0$. Let $n = n + 1$. Store the required information at node n on the right branch. Continue to Step 4.
- Step 4 On the left branch, set the tool variables of those tools required by the selected part-type to one. Check the tool magazine capacity considering the tools already loaded. If there is sufficient capacity update the tool magazine capacity and go to Step 2. Otherwise go to Step 5.
- Step 5 Fathom that node. Backtrack to node n . If $n = 0$ STOP, current solution is optimal. Otherwise, go to Step 1.

4.2.2.3. An informal description of algorithm ALG3

In algorithm ALG3, we use the same branching and fathoming rules as in algorithm ALG2. The difference between these two algorithms is in the methods of solving the subproblems. In ALG3 we solve the Lagrangian problem RSM2, (see Section 4.2) using algorithm NET and apply subgradient optimization to maximize the dual. This provides a lower bound to problem SM2 which is the LP relaxation of problem M2. Since the relaxed problem is linear, it is possible to obtain very "tight" bounds. However, the Lagrangian problem does not always provide a feasible solution to problem SM2. In algorithm ALG2 when the solution to subproblem SM2 at any node turns out to be feasible with respect to the tool magazine capacity, we obtain a solution to the original problem. But in algorithm ALG3, when we find a solution feasible with respect to the tool magazine capacity, it might not be feasible with respect to the machine time capacity. In such a case, we need a heuristic procedure to obtain a feasible solution to the original problem in order to obtain an upper bound. Hence, in algorithm ALG3, the upper bounds are obtained by the heuristic procedure HP, presented in Section 3.4. Consequently when the algorithm terminates, it provides us with a lower bound on the optimal solution value, which is a tight bound due to the linear nature of the subproblems SM2. It also provides an upper bound on the optimal solution value. This upper bound does not guarantee any error percentage. However, since the algorithm terminates with a "tight" lower bound on the exact solution value, it is possible to evaluate the quality of the heuristic upper bounds. Our motivation for designing algorithm ALG3 lies in the

fact that, the subproblems SM2 may grow large to be efficiently solved by a linear programming algorithm. In the literature, it has been reported for problems of similar structure that Lagrangian relaxation based subgradient optimization performs computationally more efficient than linear programming algorithms [Venkataramanan et. al. 1989]. Therefore, in solving large scale problems algorithm ALG3 is expected to be computationally more efficient than algorithm ALG2. However, as it will be discussed later in Chapter 5 that, for the problem sizes we have experimented with, algorithm ALG2 performed better than algorithm ALG3.

4.2.2.4. A formal statement of algorithm ALG3

- Step 0 Set $n = 0$ (number of the stored node). Set an upper bound.
- Step 1 Solve the Lagrangian problem using algorithm NET and subgradient optimization. If the lower bound is greater than the upper bound go to Step 6.
- Step 2 a. Check the feasibility of the tool magazine constraint. If the solution is feasible go Step 2b. Otherwise set $t' = 1$ and continue to Step 3.
- b. Check the feasibility of the machine time constraint. If the solution is infeasible find a feasible solution using the heuristic procedure HP. Update the upper bound and the incumbent solution if necessary and go to Step 6.
- Step 3 a. Select a variable X_{it} and which has a positive value in the optimal solution obtained in Step 1. If all X_{it} variables with positive solution values have been branched on go to Step 3b.
- b. Set $t' = t' + 1$. If $t' = T + 1$ go to Step 6. Otherwise go to Step 3a.
- Step 4 Branch on the selected variable. At the left branch set $X_{it} = X_{it}^*$ and at the

right branch set $X_n = 0$. Let $n = n+1$. Store the required information at node n on the right branch. Continue to Step 5.

- Step 5 On the left branch, set the tool variables of those tools required by the selected part-type to one. Check the tool magazine capacity considering the tools already loaded. If there is sufficient capacity update the tool magazine capacity and go to Step 3. Otherwise go to Step 6.
- Step 6 Fathom that node. Backtrack to node n . If $n = 0$ STOP, current solution is optimal. Otherwise, go to Step 1.

4.3. Model 3

The multi-machine, multi-operation model, IM, we present in Section 3.3 has the most complex structure among all models. In order to find "tight" lower bounds for problem IM in the development of an efficient solution procedure, we have studied its several relaxations. These are based on the Lagrangian Relaxation technique discussed in Geoffrion [1974] and the Lagrangian Decomposition approach presented in Guignard and Kim [1987]. However, due to the considerably complex structure of the problem, the subproblems that remain after relaxing some complicating constraints are still not easy to solve. Our purpose was to design a multiplier adjustment method which converges to a satisfactory solution in a few steps. Since the subproblems that will be solved for each multiplier update require considerable computational effort, subgradient optimization or a multiplier adjustment scheme which does not converge fast, would not be an efficient approach. However, our extensive studies on various relaxations of problem IM did not render satisfactory results. Multiplier adjustment schemes we

experimented with did not always provide "tight" bounds. With some problem instances, we were able to get lower bounds better than those provided by the LP relaxation of the problem, however, these were very data dependent. In this section, we present some relaxations of problem IM based on Lagrangian Relaxation or Decomposition techniques.

Relaxation 1: In problem IM, if we dualize constraints (18) and (20) using multipliers u_{it} (unrestricted) and w_{ijt} (nonnegative) respectively, we obtain the following three subproblems.

(LR1)

$$\begin{aligned} \text{Min } & \sum_t \sum_i \left[h_{it} I_{it}^+ + b_{it} I_{it}^- + \left(\sum_k u_{ikt} \right) X_{it} \right] \\ \text{s.t. } & I_{it-1}^+ - I_{it-1}^- + X_{it}^+ - I_{it}^+ = d_{it} \quad \forall i, t \\ & X_{it} \geq 0, \quad I_{it}^+ \geq 0, \quad I_{it}^- \geq 0 \quad \forall i, t \end{aligned}$$

(LR2)

$$\begin{aligned} \text{Min } & \sum_i \sum_{k,j} \sum_t \left(-u_{ikt} + \sum_{l \in L(i,k)} p_{lk} w_{ljt} \right) X_{ikjt} \\ \text{s.t. } & \sum_i \sum_k p_{ik} X_{ikjt} \leq C_{jt} \quad \forall j, t \\ & X_{ikjt} \geq 0 \quad \forall i, k, j, t \end{aligned}$$

(LR3)

$$\begin{aligned} \text{Min } & - \sum_i \sum_j \sum_t (w_{ijt} C_{jt}) y_{ijt} \\ \text{s.t. } & \sum_i s_i y_{ijt} \leq M_{jt} \quad \forall j, t \\ & y_{ijt} \in \{0,1\} \quad \forall i, j, t \end{aligned}$$

Problem LR1 has a network structure and problem LR2 is a continuous knapsack problem. Although these two subproblems are easy to solve, problem LR3 is still an integer problem. To obtain fast lower bounds to problem IM, we need to solve subproblems LR1, LR2, and LR3 in a multiplier adjustment scheme which converges fast (in a few multiplier updates) to the dual optimal. However, we were unable to obtain lower bounds that are better than LP relaxation in such a one or two pass algorithm. Employing subgradient optimization does not seem to be a promising alternative since, the above subproblems have to be solved at each node of a branch and bound tree in an exact solution procedure. The computational effort required in such a scheme would be beyond affordable limits.

Relaxation 2: If we eliminate the inventory variables from problem IM by substitution, the objective function and the demand constraints (17) take the following form:

$$\begin{aligned} \text{Min } & \sum_t \sum_i \left[\sum_{i=t}^T h_{it} \right] X_{it} + (h_{it} + b_{it}) I_{it}^- - \left[\sum_{i=t}^T h_{it} \right] d_{it} \\ & \sum_{i=1}^t X_{it} + I_{it}^- \geq \sum_{i=1}^t d_{it} \quad \forall i, t \end{aligned}$$

Dualizing the demand constraints and constraints (19) using multipliers v_{it} and u_{it} respectively, we obtain the relaxation of problem IM given below. Setting $v_{it} = h_{it} + b_{it}$ and adjusting u_{it} 's in a manner to improve the lower bound, we obtained bounds better than the LP relaxation bound for some problem instances. However, it does not always perform satisfactorily. Since, the relaxed problem is still an integer problem, solving it optimally requires some branch and bound scheme which has to be employed for each

adjustment of the multipliers. This is as demanding as solving the original problem in terms of the required computational effort.

$$\begin{aligned}
 & \text{Min} \sum_i \sum_t \left[\left(\sum_{l=t}^T h_{il} - v_{il} \right) X_{it} + (h_{it} + b_{it} - v_{it}) I_{it}^- - \left(\sum_{l=t}^T v_{il} - h_{il} \right) d_{it} \right] + \sum_i \sum_j \left(u_{jt} \sum_{i,k} p_{ik} X_{ikjt} - u_{jt} C_{jt} \right) \\
 & st \\
 & \quad \sum_j X_{ikjt} - X_{it} = 0 \quad \forall i, k, t \\
 & \quad \sum_{(i,k) \in L(t)} p_{ik} X_{ikjt} - C_{jt} y_{jt} \leq 0 \quad \forall j, t \\
 & \quad \sum_t s_t y_{jt} \leq M_{jt} \quad \forall t \\
 & \quad X_{it} \geq 0, \quad I_{it}^- \geq 0 \quad \forall i, t \\
 & \quad X_{ikjt} \geq 0 \quad \forall i, k, j, t \\
 & \quad y_{it} \in \{0,1\} \quad \forall i, t
 \end{aligned}$$

Relaxation 3: In problem IM, if we create duplicate copies, X'_{ikjt} , of variables X_{ikjt} for all i, k, j, t , and set them equal, we get the following constraint to be added to problem IM.

$$X_{ikjt} = X'_{ikjt} \text{ for all } i, k, j, t.$$

Relaxing this constraint and the demand constraint using multipliers w_{ikjt} (unrestricted) and v_{it} (nonnegative), we obtain the following two subproblems. Note that, inventory variables are eliminated from the formulation as in Relaxation 2.

(LD1)

$$\begin{aligned}
 & \text{Min} \sum_i \sum_t \left[\left(\sum_{l=t}^T h_{il} \right) X_{it} + (h_{it} + b_{it}) I_{it}^- - \left(\sum_{l=t}^T h_{il} \right) d_{it} + \sum_k \sum_j w_{ikjt} X'_{ikjt} \right] \\
 & st \\
 & \quad \sum_j X'_{ikjt} - X_{it} = 0 \quad \forall i, k, t \\
 & \quad \sum_i \sum_k p_{ik} X'_{ikjt} \leq C_{jt} \quad \forall j, t \\
 & \quad X_{it} \geq 0, \quad X'_{ikjt} \geq 0, \quad I_{it}^- \geq 0 \quad \forall i, j, k, t
 \end{aligned}$$

(LD2)

$$\begin{aligned}
& \text{Min} - \sum_t \sum_i \sum_k \sum_j w_{ijk} X_{ijt} \\
& \text{s.t.} \sum_{(i,k) \in L(j)} p_{ik} X_{ijk} - C_j y_{jt} \leq 0 \quad \forall j, t \\
& \sum_i s_i y_{it} \leq M_j \quad \forall j, t \\
& X_{ijk} \in \{0,1\} \quad \forall i, k, j, t \\
& y_{jt} \in \{0,1\} \quad \forall j, t
\end{aligned}$$

Relaxation 3 is similar to Relaxation 2, except that constraint (19) is not relaxed. Instead, by the use of duplicate variables the problem is decomposed into two subproblems. It is shown in Guignard and Kim [1987] that, the Lagrangian Decomposition does as well as Lagrangian Relaxation. Furthermore, if the relaxed problem does not have the integrality property then it might do better. However, for problem IM, the bounds we obtained from Relaxation 3 by a one or two step multiplier adjustment scheme were not better than those obtained from Relaxation 2.

CHAPTER 5 COMPUTATIONAL RESULTS

5.1. Model 1

5.1.1. Computational Results for Algorithm ALG1

Problem M: The branch and bound algorithm for problem M1 was programmed in C and run on an IBM PC 80386. The parameters that determine how fast the algorithm terminates are

1. Problem size determined by the values of N , L , and T ,
2. Tightness of the tool magazine constraint,
3. Tightness of the machine time constraint,
4. Error tolerance.

We have experimented with different problem sizes. In order to generate reasonably tight problems we set the tool magazine capacity to $r\%$ of the total number of tool slots required by all tools, where r is set to a value in the range $[50, 60]$. The value of r controls the tightness of the tool magazine capacity constraint. Decreasing its value, when all the other parameters are constant makes the constraint tighter. When the number of tools (L) is increased the total number of tool slots required also increases. This, in turn, increases the tool magazine capacity. However, if r is not changed, the tightness of the tool magazine capacity constraint will stay the same. The probability that a part-type uses a tool is adjusted so that the expected number of tools required by a part-

type is in the range [15,20]. The tightness of the tool magazine capacity is also determined by this parameter. The machine time capacities are set 960 minutes so that the periods are two 8-hour shifts. The period lengths should be chosen appropriately to provide that the machine time capacity constraints are not tighter than the tool magazine capacity constraints. Because, in such a case, tool magazine restrictions would become redundant and LP relaxation of the problem would be optimal.

The error tolerance is set to a value in the range [0.00,0.05]. The algorithm stops with an optimal solution within the specified error tolerance, when the error tolerance becomes greater than or equal to $[\text{Upper Bound} - \text{Lower Bound}] / \text{Upper Bound}$. If an optimal solution is not found after evaluating 60,000 nodes the algorithm is terminated.

Other problem parameters are generated randomly as follows:

Demand: Uniform between 1 and 10 units.

Processing time: Uniform between 8 and 30 minutes.

Objective function weights: Uniform between 1 and 10.

Tool slot requirement per tool: Uniform between 1 and 3 slots.

Table 2 shows the values of the chosen parameters and the corresponding problem sizes for 10 problem sets. For each set 10 problems are generated randomly. The results of the computational experiments are summarized in Table 3.

Table 2. Problem parameters

Prb Set	N	L	T	Tool Mag. Tightness	Prob. of Tool Usage	Machine Time Capacity	Error Tolerance*	No. of continuous variables	No. of binary variables
1	20	80	5	60%	25%	960	5%	100	300
2	20	60	8	60%	25%	960	1%	100	300
3	20	60	5	60%	25%	960	0%	100	300
4	50	80	8	60%	25%	960	5%	250	400
5	50	80	8	60%	25%	960	5%	250	300
6	50	80	8	60%	25%	960	0%	100	640
7	20	100	8	60%	25%	960	5%	560	800
8	70	100	10	60%	25%	960	5%	700	1000
9	70	100	10	60%	25%	960	0%	700	1000
10	70	100	10	55%	25%	960	5%	700	1000

*Error tolerance $\geq [(Lower\ Bound - Upper\ Bound) / Upper\ Bound] \times 100$

Table 3. Computational results for problem M1

Prb. Set	No. of Problems Solved	Average List Size	Average no. of Nodes	Average no. of Knapsack Problems Solved	Average CPU Seconds	Range CPU Seconds
1	10	14.2	322.6	645.0	4.51	[0.27,28.41]
2	10	16	4162.0	7228.4	62.11	[0.22,565.66]
3	9	16.6	1848.8	2510.3	26.79	[0.77,80.71]
4	9	14.2	2432	5060.1	75.15	[4.62,358.30]
5	10	18.1	190.2	576.5	6.60	[0.82,20.27]
6	9	21	2037.8	4659.1	72.45	[0.82,264.01]
7	7	22.4	237.8	851.3	12.36	[2.80,22.42]
8	10	20.8	160	599.2	7.90	[1.70,15.98]
9	10	23.1	1628.6	4762.7	85.32	[2.14,226.15]
10	6	20.3	264.66	1053.83	12.91	[2.03,29.29]

The second column in Table 3 shows the number of problems that could be solved within the specified error tolerance. As mentioned before, there are 10 randomly generated problems in each set. However, some of these may not be solved either because the computer runs out of memory or because the upper limit on the number of nodes is reached. When this happened, we disregarded those problems and took all the averages over the problems that could be solved. The third column shows the average maximum number of unfathomed nodes. The average total number of nodes opened are shown in the forth column.

It can be observed from Table 3 that problem becomes difficult to solve as N and L are increased. With $N = 20$, $L = 60$ and $T = 5$, the average execution time is 4.51 seconds (problem set 1). Making $N = 50$, $L = 80$ and $T = 6$ while all other parameters are constant, increases the execution time to 75.15 seconds (problem set 4). A similar observation is true for problem sets 5 and 7. In these problem sets, N and L are increased while T and the other problem parameters are kept constant. This increased the average execution time from 6.60 to 12.36 seconds. With problem set 7, only seven out of ten problems could be solved. In three problems instances, the program terminated due to memory restrictions and the results are disregarded. If these problems could have been solved, the average execution time would obviously be higher.

However, problem size alone does not determine how fast the algorithm terminates. Keeping N and L fixed while increasing T increases the size of the problem. But it also relaxes the problem and it becomes easier to find a feasible and optimal solution. This can be observed from problem sets 5 and 6, where the execution time

decreases from 75.15 to 6.60 seconds when T is increased from 6 to 8. This can also be observed from problem sets 7 and 8.

The tightness of the tool magazine constraint is another factor that affects termination. Setting the tool magazine capacities equal to 60% of the total number tool slots required by all tools, while the expected number of tools required by a part-type is in the range [15,25], provides reasonably tight problems. When the machine time capacity is set to 960, there is a good balance between the two capacity constraints in the sense that neither of them are loose or too tight. In problem sets 8 and 10, it is seen that decreasing the tool magazine capacity to 55% of the total number of tool slots required by all tools, increases the average execution time from 30.63 to 35.23. Also note that, in problem set 10, four problems could not be solved again due to lack of computer memory.

We have also experimented with different error tolerances. All the problems that could be solved with an error tolerance of 5% could also be solved with a zero error. Naturally this increases the execution times. However, we believe that the algorithm performs very efficiently in finding exact solutions.

Among all the problems we experimented with, only in three instances the program stopped prematurely due to the limit on the number of nodes. These are the unsolved problems in problem sets 3, 4 and 6. All other instances which were disregarded are those that could not be solved due to computer memory restrictions. The availability of memory and its effective usage is dependent on the computer environment and also the compiler. In a more effectively managed computing

environment, memory availability will be much less of a restriction for the problem sizes we are dealing with. We believe that the problems that have been solved are of real size. Also the algorithm proved to be very efficient in solving these problems with 300 to 1000 binary variables exactly.

Problem GM1: The branch and bound algorithm for problem GM1 was also programmed in C and run on an IBM PC 80386. Naturally, the problem sizes, in terms of the problem parameters, that can be solved efficiently on the PC for problem GM1 are smaller than those for problem M1. Therefore, for problem GM1, we decided to conduct some more experiments also on IBM 3090 Model 600J main frame computer. Since the size of a problem is not the only factor that determines how fast the algorithm terminates, it is not possible to give exact problem sizes that can be solved on a PC. As a matter of fact, the main restriction we had on the PC was the 64K bytes limit on the amount of memory that can be referenced. The program makes dynamic memory allocation, and as the number of unfathomed nodes and the amount of information stored at each unfathomed node reaches a certain amount, it may run out of memory. However, this limit on the memory can be overcome by using a more efficient C compiler and is not an indicator of the efficiency of the algorithm. Actually, the problems that could not be solved on PC due to this limitation were solved very efficiently on the main frame.

We set the problem parameters similarly as we did for problem M1. The tool magazine capacities are $r\%$ of the total number of tool slots required by all tools where r is set a value in the range [50,60]. The probability that a part-type uses a tool is

adjusted so that the expected number of tools required by a part-type is in the range [18,20]. As in problem M1, these two parameters have a significant role in determining the difficulty of the problem. The balance between the tool magazine capacity and machine time capacity constraints is best achieved when machine time capacity is set to either 480 or 960 minutes. Error tolerance is in the range [1%, 10%]. The limit on the number of nodes is 40000 on the PC and 240000 on the main frame. All other data are randomly generated in the same manner as in problem M1. We have generated 6 problem sets to be solved on the PC. There are 10 problems in the first four sets and 20 problems in last two. In reporting the statistics, we have disregarded those problems in which the computer runs out of memory or the upper limit on the number of nodes is reached. For the main frame (MF) we generated 6 problem sets with 10 problems in each set. On the main frame no memory problems were encountered. The problems that are disregarded are those in which the upper limit on the number of nodes is reached. Table 4 shows the chosen problem parameters for each set, problem sizes and also the computer on which the problems are tested. The results of the computational experiments are summarized in Table 5.

By observing the results summarized in Table 5, similar conclusions can be made for the algorithm's efficiency in solving problem GM1 as in solving problem M1. From the problem sets 2 and 3, we see that increasing the error tolerance from 1% to 5% causes a considerable decrease in the execution time. Making the tool magazine constraints tighter also increases the execution time as it can be observed from problem sets 3 and 4. In problem sets 5 and 6, the size of the problem increases considerably.

Table 4. Problem parameters

Prb Set	Comp Used	N	L	T	M	Tool Mag. Tight.	Prob. of Tool Usage	Machine Time Capacity	Error Toler.*	No. of cont. var.	No. of bin. var.
1	PC	20	60	5	3	60%	30%	480	1%	300	900
2	PC	20	60	5	5	60%	30%	480	1%	500	1500
3	PC	20	60	5	5	60%	30%	480	5%	500	1500
4	PC	20	60	5	5	55%	30%	480	5%	500	1500
5	PC	30	80	7	5	60%	25%	480	5%	1050	2800
6	PC	30	80	8	5	58%	23%	480	5%	1200	3200
7	MF	30	60	5	5	60%	30%	480	5%	750	1500
8	MF	30	80	5	5	60%	25%	480	5%	750	2000
9	MF	50	80	5	5	60%	25%	480	5%	1250	2000
10	MF	50	80	5	5	60%	25%	480	1%	1250	2000
11	MF	70	100	5	5	60%	20%	960	5%	1750	2500
12	MF	80	120	5	6	60%	20%	960	5%	2400	3600

* Error Tolerance $\geq [(Lower\ bound - Upper\ bound) / Upper\ bound] \times 100$

However, the average execution times are lower than that of the third problem set. This is because the problems generated in problem sets 5 and 6 are somewhat looser than those in problem set 3 due to the corresponding values of T. Same problems with smaller T values or tighter tool magazine capacities cannot be solved on PC due to the memory insufficiency. Therefore, we set T = 7 in problem set 5 and T = 8 in problem set 6 to obtain somewhat looser problems to avoid any memory insufficiency. With these parameters, some of the generated problems gave optimal solutions at the initial node which we discarded in calculating the average values. Still, there are some problem instances which cannot be solved. A general observation is that when the list size, that

Table 5. Computational Results

Prb. Set	No. of Problems Solved	Average List Size	Average no. of Nodes	Average no. of Knapsack Problems Solved	Average CPU Seconds	Range CPU Seconds
1	9	27.00	494.88	886.78	16.71	[1.70,73.35]
2	8	30.36	1871.75	4590.25	82.82	[3.19,461.04]
3	10	29.50	396.00	958.80	17.76	[2.69,143.24]
4	8	28.38	88.26	185.75	4.39	[3.62,5.00]
5	15	21.00	44.80	145.70	3.87	[0.77,6.15]
6	16	18.28	40.58	153.58	5.28	[1.15,7.64]
7	9	45.00	130.22	230.56	0.46	[0.30,0.57]
8	10	36.80	86.40	171.00	0.28	[0.09,0.51]
9	10	44.00	108.20	213.10	0.45	[0.23,0.67]
10	10	46.00	872.40	2061.70	3.10	[0.23,19.28]
11	8	72.86	229.50	425.13	2.17	[1.04,3.75]
12	9	90.33	288.44	512.67	5.63	[2.35,8.33]

is the number of unfathomed nodes becomes greater than 40, the problem becomes unsolvable on the PC due to the memory restrictions. Therefore we decided to continue our computational experiments on the main frame with much tighter problems without having any memory insufficiency.

On the main frame, we were able to solve much tighter problems which are also larger in size. In these problems we tried to keep the problem tightnesses almost the same. In problem sets 9 and 10 setting the machine time capacities to 960 provided a better balance between the tool magazine and machine time capacity constraints. As before, results show that for a given problem tightness, execution time increases as a function of the problem size. We were able to solve large MIP problems of 1500 to

3600 binary variables in very reasonable amount of CPU times within 5% to 10% of optimality. These results show the efficiency of the algorithm also in solving problem GM1.

5.1.2. Heuristic Procedure

The heuristic procedure, HP, presented in Section 3.4, is designed for the inclusive model IM, (see Section 3.3) in which the machine loading problem considers the operations of the part-types. There are also inventory holding and backordering decisions. In order to test its performance, we adapted the HP for problems M1 and GM1. In our experiments with HP, we used the same problems generated for testing the performance of algorithm ALG1. We then compared the heuristic solutions with the solutions algorithm ALG1 provided to the same problems. For problem M1, we used the problems in problem sets 3, 4, 6, 9 given in Table 2. Algorithm ALG1 provided exact solutions to the problems in problem sets 3, 6, and 9. The problems in problem set 4 were solved within a 5% error tolerance. However, since we do not have their exact solutions we used these approximate solutions in our comparisons. The results are given in Tables 6 and 7.

As it can be seen from Tables 6 and 7, HP yields solutions to problem M1 with an average error percentage in the range [2.5%, 6.8%]. In selecting the part-types, we tried the three priority rules given in the statement of HP. Instead of the backordering costs we used the objective function weights of the part-types. We then chose the best solution. Among the three priority rules, the rule which selects the part-types in a nonincreasing order of their objective function weights over processing times provided the best solution most of the time.

Table 6. Results of the Heuristic Procedure for Problem M1

Prob	Problem Set 3			Problem Set 4		
No	Opt. Val	Upp.Bnd	Error	Opt.Val	Upp Bnd	Error
1	-26108	-22756	0.026	-62175	-62175	0.000
2	-27978	-25856	0.076	-53647	-49449	0.089
3	-225644	-24156	0.058	-59310	-56464	0.090
4	-24530	-23296	0.055	-59824	-53382	0.108
5	-223001	-22704	0.013	-58980	-48408	0.089
6	-27769	-24156	0.026	-64983	-62225	0.042
7	-28620	-27060	0.055	-58980	-53324	0.096
8	-28358	-25004	0.118	-56861	-49935	0.122
9	-30559	-27748	0.092	-65680	-59746	0.090
10	-25693	-24100	0.062	-61154	-61154	0.000
Average Error :			0.0678	Average Error : 0.0673		

In order to apply HP to problem GM1, we used the problems in problem sets 2, 4, 7, 9, given in Table 4. These problems were generated to test the performance of ALG1 when applied to problem GM1. The solutions algorithm ALG1 provided to these problems are within a specified error tolerance as given in Table 4. We have compared these solutions with the solutions HP yields. As it can be observed from tables 8 and 9, for some problems in problem set 4 and 7 and for all problems in problem set 9, HP provided better solutions than ALG1. For these problem sets ALG1 yields solutions within a 5% error tolerance. This suggests that HP provides solutions with strictly less

Table 7. Results of the Heuristic Procedure for problem M1

Prob	Problem Set 6			Problem Set 9		
No	Opt. Val	Upp.Bnd	Error	Opt. Val	Upp Bnd	Error
1	-160623	-152964	0.048	-314733	-312552	0.001
2	-144293	-143878	0.003	-408570	-397557	0.027
3	-181372	-175007	0.035	-295561	-279783	0.053
4	-166685	-161448	0.048	-332411	-313713	0.056
5	-151309	-148603	0.048	-372600	-361890	0.027
6	-154715	-147308	0.048	-337845	-336375	0.009
7	-146630	-142156	0.048	-293255	-290700	0.009
8	-154164	-142107	0.078	-373398	-372816	0.001
9	-159221	-159117	0.001	-346737	-331434	0.044
10	-174341	-171752	0.015	-307415	-301914	0.018
Average Error :			0.0308	Average Error : 0.0248		

than 5% deviation from the optimal. Since HP yields better solutions for all the problems in problem set 9, we compared these solutions with the optimal LP solutions. On the average, HP results are within 1.78% of the LP optimum. In general, the average error values suggest that HP find "good" feasible solutions to problem GM1 with a negligible computational effort. For all the problems, HP provided solutions in less than a second on an IBM PC 80386.

Table 8. Results of the Heuristic Procedure for problem GM1

Prob	Problem Set 2			Problem Set 4		
No	Opt. Val	Upp.Bnd	Error	Opt. Val	Upp Bnd	Error
1	-35833	-35828	0.006	-145056	-144048	0.007
1	-43108	-42668	0.010	-146491	-146544	*
3	-40125	-39268	0.021	-159602	-159480	0.007
4	-36281	-35948	0.009	-164612	-163836	0.005
5	-35022	-34380	0.018	-134050	-134050	0.006
6	-41641	-40896	0.021	-161973	-163836	0.007
7	-42216	-40896	0.009	-165854	-166278	*
5	-40628	-40480	0.004	-147725	-147725	0.000
5	-43475	-43284	0.004	-130541	-129546	0.008
10	-42625	-42380	0.006	-144599	-163836	0.006
Average Error :			0.0102	Average Error : 0.0043		

* : HP yields a better solution than ALG1

5.2. Model 2

The branch and bound algorithms ALG2 and ALG3 presented in Section 4.2.2 were programmed in C. Algorithm ALG2 requires an LP subroutine in the solution of the lower bounding subproblems. For this reason it has been implemented on IBM 3090 Model 600J Main Frame (MF) computer which provides an Optimization Subroutine Library (OSL). The C program for ALG2 calls OSL at the nodes of the branch and bound tree where it requires to solve a lower bounding subproblem. ALG3 uses

Table 9. Results of the Heuristic Procedure for problem GM1

Prob	Problem Set 7			Problem Set 9		
No	Opt. Val	Upp.Bnd	Error	Opt.Val / LP Opt.	Upp Bnd	Error
3	-75960	-75701	0.003	-110697 / -115938	-115337	0.005
2	-67050	-67362	*	-110857 / -116312	-115540	0.007
3	-74900	-73632	0.017	-119637 / -123860	-119666	0.033
4	-80523	-80208	0.004	-117345 / -121243	-120527	0.006
5	-73971	-74467	*	-118317 / -123286	-120628	0.022
3	-80329	-79009	0.016	-109930 / -115762	-111989	0.033
7	-75960	-74119	0.024	-104267 / -109805	-105744	0.037
8	-62782	-61798	0.016	-106748 / -111507	-110475	0.009
8	-63921	-64479	*	-105926 / -111774	-109304	0.022
10	-67281	-65915	0.020	-108011 / -111368	-111313	0.000
Average Error :			0.0143	Average Error : 0.0175		

*: HP yields a better solution than ALG1

subgradient optimization to solve the lower bounding subproblems by using a network algorithm as discussed in section 4.2.2. Since, it does not use a package program that would require a certain computing environment, it can be run on any computer. We have experimented with ALG3 on IBM PC 80386, on MF computer, and also on SUN Workstations. However, since ALG2 performs better than ALG3 for solving problem M2, we have focused our computational experiments on ALG2. Algorithm ALG2 when terminates finds an exact solution to problem M2. Algorithm ALG3 provides a lower

bound on the optimal solution value and a heuristic solution which does not guarantee any particular error percentage. Further in this section we will present a comparison of the two algorithms. We now present the results of the computational experiments for algorithm ALG2.

5.2.1 Computational Results for Algorithm ALG2

In order to test the performance of algorithm ALG2, we have randomly generated 20 problem sets with 10 problems in each set. As in algorithm ALG1, the following parameters determine how quickly the algorithm terminates.

1. Problem size as determined by the values of N , L , T , and M .
2. Problem tightness as determined by the two capacity constraints.
3. Error tolerance.

We have experimented with different problem sizes. We set the tool magazine capacity to $r\%$ of the total number of tool slots required by all tools, where r is set to a value in the range $[50, 60]$. As in algorithm ALG1, the value of r controls the tightness of the tool magazine capacity constraint. The probability that a part-type uses a tool is set to a value in the range $[0.18, 0.30]$. The tightness of the tool magazine capacity constraint is also controlled by this parameter. The machine time capacity is set to 300, 400, or 480 depending on the values of N , M , and T , so that the corresponding constraints are neither too tight nor too loose. If the machine time capacity constraints are very tight, the problems will become easy to solve. Therefore, a good balance must be established between the two capacity constraints.

Similar to algorithm ALG1, the error tolerance is set to a value in the range $[0.00, 0.05]$. The algorithm stops with an optimal solution within the specified error tolerance, when the error tolerance becomes greater than or equal to $[\text{Upper Bound} - \text{Lower Bound}] / \text{Upper Bound}$. If an optimal solution is not found after evaluating 180,000 nodes the algorithm is terminated.

Other problem parameters are generated randomly as follows:

Demand: Uniform between 0 and 10 units.

Processing time: Uniform between 8 and 30 minutes.

Inventory holding costs: Uniform between 1 and 10.

Backorder costs: 2 x inventory holding costs

Tool slot requirement per tool: Uniform between 1 and 3 slots.

Problem M2: Table 10 shows the values of the chosen parameters and the corresponding problem sizes for 12 problem sets generated for problem M2. For each set 10 problems are generated randomly.

The results of our computational experiments are summarized in Table 11. The second column in Table 11 shows the number of problems that were solved within the specified error tolerance, out of the 10 problems generated for each problem set. For some problems the algorithm terminated prematurely by reaching the upper limit on the number of nodes. The third column shows the average maximum number of unfathomed nodes. The average total number of nodes evaluated is shown in the forth column and the average number of simplex iterations performed is given in the fifth column.

Table 10. Problem Parameters for Problem M2

Prb Set	N	L	T	Tool Mag. Tightness (r%)	Prob.of tool usage	Mach. time cap.	Error Toler.*	No. of cont. var.	No. of binary var.
1	8	20	3	60%	30%	300	0.1%	72	90
2	9	30	3	50%	25%	300	0.1%	81	90
3	10	30	3	60%	25%	400	0.1%	90	90
4	10	30	5	50%	20%	300	0.1%	150	150
5	10	30	5	50%	20%	300	5%	150	150
6	10	40	5	50%	20%	300	5%	150	200
7	12	40	5	55%	20%	300	5%	180	200
8	12	50	5	60%	20%	300	5%	180	250
9	15	50	5	60%	20%	400	5%	225	250
10	15	60	5	60%	20%	400	5%	225	300
11	20	60	5	60%	20%	480	5%	300	300
12	50	80	5	60%	20%	480	5%	750	400

* Error tolerance $\geq [(\text{Upper bound} - \text{Lower bound}) / \text{Upper bound}] \times 100$

As it can be seen from Table 11, increasing N and/or L while the other problem parameters are kept constant increases the execution time. In problem sets 2 and 3, N is increased from 9 to 10 while all other problem parameters are kept constant. This caused the average execution time to increase from 5.27 to 6.39 seconds. In problems sets 5 and 6, L is increased from 30 to 40 causing an increase from 1.11 to 9.44 seconds in the average execution time. A more dramatic increase can be observed in problem sets 9 and 10 where L is increased from 50 to 60. The average execution time increased from 11.18 to 184.7 seconds.

Table 11. Computational Results for Problem M2

Prb. Set	No. of Problems Solved	Average List Size	Average no. of Nodes	Average no. of Simplex Iterations	Average CPU Seconds	Range CPU Seconds
1	10	8.5	312.4	698.9	1.40	[0.03,3.52]
2	10	12.5	1133.6	2580.1	5.27	[0.54,23.14]
4	10	14.6	1327.4	2738.6	6.39	[0.76,37.72]
4	10	17.6	1034.2	2247.9	9.44	[1.67,16.95]
5	10	16.6	169.2	458.2	1.11	[0.40,2.46]
5	10	14.6	1411.8	3928.2	9.44	[0.22,56.43]
7	10	16.6	2899.0	8333.4	21.44	[0.40,132.29]
8	10	15.9	866.2	2398.2	6.31	[0.08,50.83]
9	10	19.8	1442.8	2951.1	11.18	[0.39,72.25]
10	10	20.7	21705.2	61449.5	184.70	[0.49,631.57]
11	9	24.8	24579.3	100381.2	383.53	[4.40,1225.22]
12	6	26.0	8538.0	19729.7	165.05	[4.93,539.99]

However, size of the problem is not the only factor that determines how quickly the algorithm finds an optimal solution. The tightness of the problem determined by the two capacity constraints is also a major factor. In problem sets 3 and 4, T is increased from 3 to 5 causing an increase from 90 to 150, in the number of both the continuous and the binary variables. This is expected to cause an increase in the average execution time. However, in problem set 4, the probability that a part-type uses a tool is decreased to 20% which leads to a decrease by 3 units in the expected number of tool slots used by each part-type. Although, the capacity of the tool magazine constraint is also decreased by 10% in problem set 4, the overall effect is a looser tool magazine constraint. Therefore, the average execution times for the problem sets 3 and 4 are

almost the same. A similar argument can be made for problem sets 7 and 8 where L is increased from 40 to 50, causing an increase from 200 to 250 in the number of binary variables. Although, this is expected to increase the average execution time, since r is increased by 5, the tool magazine capacity becomes looser and this leads to a decrease from 21.44 to 6.31 seconds in the average execution time. This behavior can also be observed in problem sets 7 and 9 where both N and L increase, increasing the number of both the continuous and the binary variables. However, since r is increased by 5, the tool magazine capacity becomes looser and the average execution time decreases.

The value of the error tolerance also affects how quickly the algorithm terminates. In problem sets 4 and 5, error tolerance is changed from 0.1% to 5% while all the other problem parameters are kept constant. This caused a decrease from 6.40 to 1.11 seconds in the average execution time.

As it can be seen from the second column, 1 out of the 10 problems in problem set 11 and 4 out of the 10 problems in problem set 12 could not be solved. For these problem sets, the average values reported in Table 2 were calculated over the problems that could be solved. Consequently, the average values in problem set 12 are lower than those in problem set 11. If the statistics for unsolved problems were included, the average values for problem set 12 would have been higher. We have tried to solve the unsolved problems by increasing the error tolerance to 10%. With 10% tolerance we were able to solve 2 of the 4 unsolved problems in problem set 12.

The average maximum number of unfathomed nodes (column 3) generally increases as the problems get larger. Problem size has a more dramatic impact on the

average number of nodes in the branch and bound tree as shown in column 4. As expected, behavior of the average number of nodes and the average number of simplex iterations (column 6) parallel the behavior of the average execution time. However, the number of simplex iterations does not increase at a constant rate as the number of nodes increases. When a lower bounding problem is solved at a node of the branch and bound tree, OSL does not start from an artificial basis but uses the last optimal solution it has found as a starting solution. Therefore, the number of iterations depends on how "good" this starting solution is. Note again that these average values exclude the problems in which the algorithm stopped prematurely after opening 180,000 nodes.

Problem GM2: For problem GM2, we have experimented with 8 problem sets with 10 randomly generated problems in each set. The values of the chosen parameters and the corresponding problem sizes are given in Table 12.

Table 12. Problem Parameters for Problem GM2

Prb Set	N	L	T	M	Tool Mag. Tightness (r%)	Prb.of tool usage	Mach. time cap.	Err. Tol.	No. of cont. var.	No. of binary var.
1	10	40	5	2	60%	20%	300	5%	200	400
2	12	50	5	2	60%	20%	300	5%	240	500
3	15	50	5	2	60%	20%	400	5%	300	500
4	15	60	5	2	55%	18%	300	5%	300	600
5	20	60	5	2	60%	18%	300	5%	400	600
6	20	60	5	2	55%	18%	300	5%	400	600
7	20	60	5	2	60%	20%	300	5%	400	600
8	50	80	5	2	60%	18%	300	5%	1000	800

* Error tolerance $\geq [(Upper\ bound - Lower\ bound) / Upper\ bound] \times 100$

The results of the computational experiments are summarized in Table 13.

Table 13. Computational Results for Problem GM2

Prb. Set	No. of Problems Solved	Average List Size	Average no. of Nodes	Average no. of Simplex Iterations	Average CPU Seconds	Range CPU Seconds
1	9	33.11	14714.00	27178.8	112.11	[0.096,852.97]
2	9	35.89	15552.66	37441.3	144.09	[0.41,1131.08]
1	9	47.29	6223.60	11502.8	61.72	[0.75,242.7]
1	9	41.33	17628.22	28065.2	176.42	[0.86,1512.71]
5	10	41.60	1858.80	3541.0	22.11	[0.92,195.74]
5	9	43.44	22593.32	74969.1	307.69	[1.12,1678.67]
7	9	41.89	2168.89	8550.0	30.74	[0.50,256.8]
8	9	52.33	643.54	1732.2	16.04	[7.93,43.50]

The computational results tabulated above for problem GM2, lead to similar conclusions we have made for problem M2, about the relationship between the difficulty of the problem and the selected problem parameters.

In general the problem becomes more difficult as the problem size increases as a function of N and L . This can be observed from problem sets 1 and 2 where all average values increase as N and L are increased. It can also be seen in problem sets 5 and 8. Although the reported averages for problem set 8 are smaller than those for problem set 5, this is due to the fact that 1 out of 10 problems generated for problem set 8 could not be solved within the allowable limits and therefore is not included in the calculation of those averages. If it was included, the averages for problem set 8 would have been higher. In fact, the number of problems solved (column 2) is a good indicator of the difficulty of the problems.

In problem set 3, only 5 out of 10 the generated problems could be solved. This suggests that these are very tight problems. As mentioned before, problem tightness is dictated by the interaction of the machine time and the tool magazine capacity constraints. Problem sets 3 and 4 constitute a very good example for this argument. In problem set 4, L is increased by 10. The probability that a part-type uses a tool is decreased to 18% in order to keep the expected number of tools required by a part-type almost the same. In both problem sets 3 and 4 each part-type, on the average, requires 10 tools and 20 tool slots. However, in problem set 4, r is decreased by 5 leading to a tighter tool magazine capacity. Accordingly, problems in problem set 4 are expected to be more difficult than those in problem set 3. However, in problem set 4 the machine time capacity is also reduced by 100 units making both the machine and the tool magazine capacity constraints tighter than they are in problem set 3. Consequently, the overall effect was that the problems in set 4 became easier to solve and 9 out of the 10 problems could be solved. In problem set 3, where the tool magazine capacities are relatively more dominating than the machine time capacities, the problems are more difficult to solve. Hence, the balance between the machine time and the tool magazine capacities determine how hard or easy a problem is.

It can be seen from problem sets 5 and 6 that, decreasing the tool magazine capacity, while all other problem parameters are kept the same, causes a significant increase in all the average values. Note also that 1 out of 10 problems in problem set 6 could not be solved. Between problem sets 5 and 7, the probability that a part-type uses a tool changes by 2%, which indicates a tighter tool magazine capacity constraint in

problem set 7 and leads to higher average values. However, the increases in these averages are not as much as the increases in the averages between problem sets 5 and 6. This suggests that, tightening the tool magazine constraint by decreasing the capacity leads to a tighter constraint than tightening it by increasing the probability that a part-type uses a tool. This can also be directly observed from problem sets 6 and 7.

The computational results show that the branch and bound algorithms developed both for problems M2 and GM2 perform quite efficiently. The average execution times for problems solved within the specified error tolerance are reasonable when the number of binary variables are considered. Problems with much smaller dimensions than those considered in this paper could not be solved using a standard MIP package within a reasonable execution time.

5.2.2. Algorithm ALG3: Implementation Issues

As it has been discussed in earlier sections, the lower bounding subproblems are solved by subgradient optimization in algorithm ALG3. At the initial node of the branch and bound tree, subgradient iterations starts with values of Lagrangian multipliers set to zero. After 150 iterations, the algorithm terminates with a lower bound on the optimal value of the side constrained network problem. Our experiments show that 150 iterations is sufficient to obtain "tight" lower bounds when the iterations start with zero multiplier values. However, at the child nodes, the initial values of the multipliers can be set to the near optimal multiplier values obtained at the parent nodes. The difference between the subproblems solved at a node and at its child node is only in one variable. Therefore, the near optimal multiplier values obtained at a node will serve as good

starting values for the multipliers at its child node. Furthermore since these multiplier values are actually close to the near optimal multiplier values for the child node, the number of iterations can be decreased. Hence, at all other nodes of the branch and bound tree except the root node, we terminated the subgradient algorithm at 30 iterations. As it can be seen from Table 8 given in the next section, the lower bounds obtained are quite "tight". This supports that 30 iterations is sufficient in finding a lower bound to the subproblems at the nodes except the root node.

5.2.3. Comparison of ALG2 and ALG3

We have made comparative testings of the performance of algorithms ALG2 and ALG3. With the problem parameters set as $N = 8$, $L = 20$ and $T = 3$, we have solved the 10 randomly generated problems in problem set 1 (Table 6) by both algorithms. The results are presented in Table 14.

ALG3 terminates providing a lower and an upper bound on the optimal objective value, and a solution that gives the upper bound. This solution is obtained by a heuristic procedure and does not guarantee any limit on the percentage error. As it can be seen from the above table (column 4) the percentage error for the lower bounds is within the range of $[0\%, 7.8\%]$, with an average of 3.04%. This indicates that the subgradient algorithm provides good quality solutions for the Lagrangian subproblems. However, compared to the direct method (by a linear programming algorithm) of solving the subproblems, it does not prove to be very efficient for the size of the problems we are dealing with. The network problem with side constraints (see Section 4.2) involve $3 \times N \times T$ variables and $N \times T + T$ constraints, which does not grow large even for problems

Table 14. Comparison of ALG2 and ALG3

Prb. No	Optimal Value ALG2 (z^*)	Lower Bound ALG3 (z_L)	Error ^a	Upper Bound ALG3 (z_U)	Error ^b	Exec. Time ALG2	Exec. Time ALG3 MF	Exec. Time ALG3 Sun ^c
1	1286.57	1214.06	0.056	1316.57	0.023	0.236	3.09	2.20
2	597.17	572.89	0.041	616.50	0.032	1.975	8.09	29.30
3	506.81	489.43	0.034	510.67	0.008	0.450	9.28	10.50
4	225.45	218.53	0.031	229.00	0.016	3.151	8.09	44.90
5	429.11	396.07	0.077	432.20	0.007	2.904	9.28	48.40
6	344.33	340.06	0.012	353.00	0.025	1.186	7.70	31.90
7	1155.94	1124.01	0.028	1304.67	0.129	3.521	14.72	99.00
8	1568.92	1568.89	0.000	1586.20	0.000	0.029	0.42	1.20
9	833.10	832.03	0.001	1075.86	0.291	0.168	0.39	1.10
10	1047.51	1022.85	0.024	1047.51	0.000	1.270	5.68	31.00

(a) Error on the lower bound = $(z^* - z_L) / z^*$ (b) Error on the upper bound = $(z_U - z^*) / z^*$

(c) Execution time on Sun Workstations

of real size. Therefore, a linear programming algorithm can efficiently solve these problems. Consequently algorithm ALG2 performs more efficiently in terms of computation time than algorithm ALG3. Average computation time for algorithm ALG2 is 1.49 CPU second on the main frame while it is 5.38 CPU seconds for algorithm ALG3. The upper bounds obtained by algorithm ALG3 has a percentage error in the range [0%,29%] (column 6) with an average value of 5.31%. This value is quite good

for an approximate solution, however since algorithm ALG2 provides exact solutions with less computational effort, it outperforms algorithm ALG3.

5.2.4. Heuristic Procedure

We have experimented with the heuristic procedure, HP, presented in Section 3.4, for finding solutions to problem M2. In order to test its performance, we have used the first 4 problem sets in Table 10, which are generated to experiment with algorithm ALG2. Note that the solutions provided by algorithm ALG2 for these problems are within a 0.1% error tolerance. The following tables gives a summary of our computational experiments with HP, for problem M2. The computation times for all problems were less then a second on an IBM PC 80386.

As it can be observed from the above tables, the heuristic procedure yields solutions to problem M2 with an average error percentage in the range [8%,24%]. The overall average error percentage is 14.7%. Problem M2 is a difficult problem involving multi period decisions on production, inventory, and backorder quantities. Heuristic rules that provide good quality solutions for single period problems do not always perform as well in a multi period problem due to interactions among the periods. A decision made in one period has a significant impact over all periods. However, capturing all this interaction in a heuristic procedure might not be computationally feasible.

Table 15. Results of the Heuristic Procedure for Problem M2

Prob No	Problem Set 1			Problem Set 2		
	Opt. Val	Upp.Bnd	Error	Opt.Val	Upp Bnd	Error
1	1047.5	1169.1	0.116	861.9	811.7	0.013
2	833.1	833.1	0.000	497.7	178.9	0.139
3	1568.9	1568.9	0.000	497.7	589.3	0.155
4	1155.9	1187.1	0.026	256.1	263.6	0.029
5	344.3	369.7	0.074	597.8	962.2	0.609
6	429.1	513.2	0.196	376.6	666.6	0.770
7	225.5	259	0.196	854.2	1087.6	0.273
8	506.8	521.3	0.029	1204.7	1314.0	0.091
9	597.2	710.2	0.189	529.6	663.9	0.254
10	1286.6	1316.6	0.023	861.9	930.0	0.057
Average Error :			0.080	Average Error : 0.239		

We employ three different heuristic rules none of which is dominating in terms of the quality of the solutions they provide. In a given period part-types are selected according to a nonincreasing order of one of the following priorities:

- i. backorder cost
- ii. backorder cost / processing time
- iii. backorder cost / tool slot requirement.

We then use these rules in two different heuristic procedures. These heuristic procedures

are based on a period by period consideration of the problem which does not always

Table 16. Results of the Heuristic Procedure for Problem M2

Prob	Problem Set 3			Problem Set 4		
No	Opt. Val	Upp.Bnd	Error	Opt.Val	Upp Bnd	Error
1	816.6	862.7	0.166	2615.6	3022.7	0.156
2	309.6	401.6	0.297	1315.3	1524.4	0.156
3	1482.5	1610.5	0.086	4442.2	4573.5	0.029
4	708.5	764.1	0.078	2608.6	2704.3	0.037
5	1145.5	1395.9	0.217	4038.0	4044.0	0.001
6	671.1	1102.3	0.643	3032.3	3447.7	0.137
7	783.4	854.9	0.091	944.3	1145.5	0.213
8	1140.7	1187.2	0.041	3534.0	3536.1	0.001
9	1251.2	1459.0	0.166	4308.8	4563.3	0.059
10	441.0	495.6	0.124	3056.3	3403.5	0.114
Average Error : 0.179				Average Error : 0.091		

provide the best solution to the overall problem. The first procedure is heuristic HP as it is presented in Section 3.4. In heuristic HP, the selected part-types are produced in an amount equal to their demands in each period. Any shortage is then attempted to be produced to inventory in previous periods, if it is feasible with respect to tool magazine and machine time capacities. However, since the same selection rule is used, this leads to producing the same collection of part-types in every period. Furthermore, since they are produced in their "just-in-time" quantities in all periods, there is usually no need to

carry inventories for these part-types. Part-types which are not selected for production in the periods when they are demanded are then attempted to be produced to inventory in previous periods. However, tool magazine capacities might not allow this to be feasible even if there is some unutilized machine time capacity. In such a case it would be preferable to produce an already selected part-type more than its "just-in-time" quantity and carry inventory. This way the machine time capacity would be more effectively utilized and if all its demand is satisfied from inventory, a new collection of part-types would be selected in the next period. Although this seems to increase inventory carrying costs it might lead to less backordering which is more expensive. In order to provide this, we have modified the strategy for production to inventory in the heuristic procedure HP. In the modified procedure, a selected part-type is produced either at an amount equal to its total demand for the whole planning horizon or as much as the machine time capacity permits. Any amount produced in excess of its demand in a period is carried in inventory.

The two heuristic procedures which employ different strategies for production to inventory, are tested with each of the three priority rules for part-type selection. Therefore, we solved each problem six times and chose the best solution. Although, the first and second rules seem to perform better than the third, none of them was uniformly dominating. Similar arguments hold for the heuristic procedures.

CHAPTER 6

CONCLUSION AND FURTHER RESEARCH

In this research we investigate certain planning problems in an FMS environment. They are usually referred as the set-up problems since, they involve decisions about setting up the system before production starts. We present five MIP formulations. In these models, the problem is viewed over a T-period planning horizon. There are two chore models that deal with different demand situations. Model 1 has an objective function in the form of minimization of the total weighted flow time, which provides scheduling the processings of more urgent part-types in the earlier periods. Model 2 minimizes inventory and backorder costs over the planning horizon. For both cases, we start with a single-machine formulation for part-type selection and lot-sizing problems. Machine and tool magazine capacities are aggregated to represent overall system restrictions. Subsequently, we generalize both formulations to consider each machine separately. This incorporates machine loading problem into the models. There is no explicit consideration of operations of the part-types however, it is implicitly assumed that all operations of a part-type are assigned to the same machine. In these multi-machine models, fixture allocations are also considered by treating fixtures as special machine types. Finally, we present an inclusive model which considers operations of the part-types explicitly in machine loading. This model, Model 3, deals with the part-type

selection, lot sizing, and machine loading problems simultaneously and considers many major attributes of these problems.

There are several distinguishing features of our models. In the FMS literature, the planning problems are usually formulated as single period problems. Those researchers who present multi-period formulations do not suggest any solution procedures to obtain an optimal or near optimal solution to the problem. Single period problems can always be written as special cases of the formulations we present.

The lot-sizing issue is not considered as an integrated feature of the FMS planning models given in the literature. In the formulations for part-type selection or machine loading problems, lot sizes are either assumed to be predetermined or not considered at all. We consider lot sizing as one of the main attributes of the FMS planning problem. When a part-type is selected, its lot size and lot sizes of its operations on the machines they are assigned to are determined based on the aggregated or individual machine capacities.

In most of the research on FMS planning, there is no explicit consideration of due dates. However, since most FMS users are mainly concerned with meeting due dates we based our planning decisions on this criterion.

For Model 1, we have devised an exact solution procedure which finds optimal or near optimal solutions to both single machine and multi-machine problems. It is a branch and bound algorithm where LP relaxation of the problem is solved to obtain lower bounds. We have shown that, due to a special problem structure these lower bounding LP problems can be solved very easily without resorting to a linear programming

algorithm. It is a very efficient solution procedure. Our computational experiments show that both single and multi-machine problems of real size, which cannot be solved using standard MIP packages, are solved quite fast on an IBM PC 30386.

For Model 2, we have designed two branch and bound algorithms which differ in their methods of solving the lower bounding subproblems. Both are based on the LP relaxation of the problem. However, the relaxed problem has a different structure from that in Model 1. In one algorithm we resort to a standard package that uses a LP algorithm to solve the lower bounding LP problems. In the other algorithm, we use subgradient optimization based on Lagrangian relaxation. Based on our experiments we concluded that the algorithm which uses the standard LP package for solving the subproblems performs more efficiently. Therefore, we have experimented with this algorithm for both the single machine and the multi-machine formulations of Model 2. The algorithm performed efficiently with moderate problem sizes. Due to certain restrictions on the computing facilities, we could not experiment with problems of larger sizes. For Model 2, we have also experimented with two adaptations of a heuristic procedure which is developed to find solutions for Model 3, the most general model. These heuristic procedures find solutions with an average error of 14%.

Model 3, the most general model that considers multiple machines and multiple operations, has a very complex structure. We have studied several relaxations of the problem based on Lagrangian Relaxation and Lagrangian Decomposition techniques. However, these did not render satisfactory results in terms of the quality of the lower bounds they provide and the computational effort they require. Therefore, we have

developed a heuristic procedure. In further research, we plan to test the performance of this heuristic comparing the results it provides with the results of an exact solution procedure at least for sample problems of small size.

The branch and bound algorithms designed for Models 1 and 2 are different from each other due to the major differences in the problem structures. The lower bounding subproblems require different solution techniques. In Model 1, the LP relaxation of the problem is separable with respect to the time periods. Furthermore each separated problem has a knapsack structure. The branch and bound algorithm for Model 1 exploits these characteristics in solving the lower bounding subproblems. The computational performance of the algorithm devised for Model 1 is very satisfactory. Problems having 1500 to 3600 binary variables could be solved near optimally within a CPU time of 2.50 to 19 seconds on a main frame computer. For smaller size problems (up to 1500 binary variables) solutions can be obtained on a personal computer within 4 to 465 seconds of CPU time. In Model 2, the lower bounding subproblems do not have such an exploitable structure. Furthermore, the problem involves decisions on inventory carrying and backordering which are not incorporated in Model 1. Naturally, the algorithms designed for Model 2 required longer CPU times compared to the algorithm designed for Model 1, for problems having the same parameters. With the same problem parameters, the number of continuous variables in Model 2 is three times this number for Model 1, due to variables representing inventory and backorder quantities. Yet, the algorithm solves problems of moderate size (of 400-800 binary variables) in a CPU time of 0.096 to 1680 seconds on the main frame computer.

For further research, we plan to work on the most general model, Model 3, in which the machine loading problem considers the operations of the part-types. Our purpose is to design procedures in order to find "tight" bounds to the problem in within reasonable computation times. Furthermore, we plan to extend Model 1, also to incorporate the operations of the part-types. We believe that Lagrangian Relaxation based procedures can be developed for finding approximate solutions to this extended problem.

REFERENCES

- Afentakis, P., "Maximum Throughput in Flexible Manufacturing Systems," Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 509-172, Amsterdam, 1986.
- Afentakis, P., M.M. Solomon, and R. Millen, "The Part-Type Selection Problem," Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 141-146, Amsterdam, 1989.
- Akella, R., Y. Choong and S.B. Gershwin, "Real-Time Production Scheduling of an Automated Cardline," Annals of Operations Research, vol. 3, pp. 403-425, 1985.
- Ammons, J.C., C.B. Lofgren, and L.F. McGinnis, "A Large Scale Machine Loading Problem in Flexible Assembly," Annals of Operations Research, vol. 3, pp. 319-332, 1985.
- Bastos, M. Jose, "Batching and Routing: Two Functions in the Operational Planning of Flexible manufacturing Systems," European Journal of Operations Research, vol. 33, pp. 230-244, 1988.
- Berrada M., and Kathryn E. Stecke, "A Branch and Bound Algorithm for Machine Load Balancing in Flexible Manufacturing Systems," Management Science, vol. 32, no. 10, 1986.
- Birge J.R., "Real-Time Adaptive Scheduling in Flexible Manufacturing Systems," Technical Report 85-26, Dept. of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1985.
- Brown, J., W.W. Chan, and K. Rathmill, "An integrated FMS Design Procedure," Annals of Operations Research, vol. 3, pp. 207-237, 1985.
- Buzacott, J.A., and J.G. Shantikumar, "Models for Understanding Flexible Manufacturing Systems," AIIE Transactions, pp. 339-349, December 1980.

- Buzacott, J.A., and D.D. Yao, "Flexible Manufacturing Systems: A Review of Models," *Management Science*, vol. 32, no. 7, pp. 890-905, 1986.
- Carrie, A.S., E. Adhami, A. Stephens, and I.C. Murdoch, "Introducing a Flexible Manufacturing System," *International Journal of Production Research*, vol. 22, no. 6, pp. 907-916, 1984.
- Carrie, A.S., and D.T.S. Perera, "Work Scheduling in FMS Under Tool Availability Constraints," *International Journal of Production Research*, vol. 24, no. 6, pp. 1299-1308, 1986.
- Carrie, A.S., and A.C. Petsopoulos, "Operations Sequencing in an FMS," *Robotica*, vol. 3 p. 259, 1985.
- Chakravarty A.K., and A. Shtub, "Selecting Parts and Loading Flexible Manufacturing Systems," *Proceedings of the First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 284-289, Amsterdam, 1984.
- Chakravarty A.K., and A. Shtub, "Production Planning with Flexibilities in Capacity," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 333-343, Amsterdam, 1986.
- Chang, Y.L., and R.S. Sullivan, "Lot Sizing in Flexible Assembly Systems," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 360-368, Amsterdam, 1986.
- Chang, Y.L., R.S. Sullivan, U. Bagchi, and J.R. Wilson, "Experimental Investigation of Real-Time Scheduling in Flexible Manufacturing Systems," *Annals of Operations Research*, vol. 3, pp. 355-377, 1985.
- Chung, S.H., and G.L. Doong, "A Procedure to Solve Part Mix and Tool Assignment Problems in FMS," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publ. B.V., pp. 111-116, Amsterdam, 1989.
- Chung, S.H., and T.R. Lee, "A Heuristic Method for Solving FMS Master Production Scheduling Problem," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 127-132, Amsterdam, 1989.

- Dallery, Y., and Y. Frein, "An Efficient Method to Determine the Optimal Configuration of a Manufacturing System," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 360-368, Amsterdam, 1986.
- Dupont-Gatelmand, C., "A Survey of Flexible Manufacturing Systems," *Journal of Manufacturing Systems*, vol. 1, no. 1, 1985.
- Erenguc, S.S., and S. Tufekci, "A Transportation Type Aggregate Production Model with Bounds on Inventory and Backordering," *European Journal of Operational Research*, vol. 35, pp. 414-425, 1988.
- Erschler, J., F. Roubellat, and C. Thuriot, "Steady State Scheduling of a Flexible Manufacturing System with Periodic Releasing and Flow Time," *Annals of Operations Research*, vol. 3, pp. 333-353, 1985.
- Faaland, B., "A Weighted Selection Algorithm for Certain Tree-Structured Linear Programs," *Operations Research*, vol. 32, no. 2, 1984.
- Garey, M.R., and D.S. Johnson, *Computers and Intractability*, W. H. Freeman and Company, San Francisco, 1979.
- Geoffrion, A.M., "Lagrangian Relaxation for Integer Programming," *Mathematical Programming Study 2*, North-Holland, pp. 82-114, Amsterdam, 1974.
- Greene, T.J., and R.P. Sadowski, "A Mixed Integer Program for Loading and Scheduling Multiple Flexible Manufacturing Cells," *European Journal of Operational Research*, vol. 24, pp. 379-386, 1986.
- Guignard, M., and S. Kim, "Lagrangian Decomposition: A Model Yielding Stronger Lagrangian Bounds," *Mathematical Programming*, vol. 39, pp. 215-228, 1987.
- Han, M.-H., and L.F. McGinnis, "Flow Control in Flexible Manufacturing: Minimization of Stockout Cost," *International Journal of Production Research*, vol. 27, no. 4, pp. 701-715, 1989.
- Han, M.-H., Y.K. Na, and G.L. Hogg, "Real-Time Tool Control and Job Dispatching in Flexible Manufacturing Systems," *International Journal of Production Research*, vol. 27, no. 8, pp. 1257-1267, 1989.
- Heungsoon, F.L., M.M. Srinivasan, and A.C. Yano, "An Algorithm for the Minimum Cost Configuration Problem in Flexible Manufacturing Systems," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems*:

Operations Research Models and Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 127-132, Amsterdam, 1989.

- Hirabayashi, R., H. Suzuki, and N. Tsuchiya, "Optimal Tool Module Design Problem for NC Machine Tools," *Journal of the Operations Research Society of Japan*, vol. 27, no. 3, pp. 205-229, 1984.
- Hitz, K.L., "Scheduling of Flexible Flow Shops I," LIDS Report No. 879, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1979.
- Hitz, K.L., "Scheduling of Flexible Flow Shops II," LIDS Report No. 1049, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1980.
- Huang, P.Y., and C.-S. Chen, "Flexible Manufacturing Systems: An Overview and Bibliography," *Production and Inventory Management*, Third Quarter, 1986.
- Hutchison, J., K. Leong, D. Snyder, and F. Ward, "Scheduling for Random Job Shop Flexible Manufacturing Systems," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 127-132, Amsterdam, 1989.
- Hwang, S., "A Constraint Directed Method to Solve the Part-Type selection Problem in Flexible Manufacturing Systems," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 297-309, Amsterdam, 1986.
- Hwang, S., and A.W. Shogan, "Modelling and Solving an FMS Part-Type Selection Problem," *International Journal of Production Research*, vol. 27, no. 8, pp. 1349-1366, 1989.
- Jaikumar, R., "Postindustrial Manufacturing," *Harvard Business Review*, pp. 69-76, Nov-Dec 1986.
- Jaikumar, R., and L.N.V. Wassenhove, "A Production Planning Framework for Flexible Manufacturing Systems," *Harvard Business School Working Paper*, 1987.
- Kalkunte, M.V., S.C. Sarin, and W.E. Wilhelm, "Flexible Manufacturing Systems: A Review of Modeling Approaches for Design, Justification and Operation," in *Flexible Manufacturing Systems: Methods and Studies*, A. Kusiak (Editor), Elsevier Science Publishers B.V., North-Holland, pp. 3-25, Amsterdam, 1986.

- Kennington J.L., and R.V. Helgason, *Algorithms for Network Programming*, John Wiley & Sons, New York, 1980.
- Kim, Y.-D., and C.A. Yano, "A Heuristic Approach for Loading Problems of Flexible Manufacturing Systems," Technical Report 87-21, Dept. of industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1987a.
- Kim, Y.-D., and C.A. Yano, "A Branch and Bound Approach for the Loading Problem in Flexible Manufacturing Systems: An Unbalancing Case," Technical Report 87-18, Dept. of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1987b.
- Kim, Y.-D., and C.A. Yano, "A New Branch and Bound Approach for Loading Problems in Flexible Manufacturing Systems," Technical Report 89-5, Dept. of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1989a.
- Kim, Y.-D., and C.A. Yano, "An Iterative Approach to System Setup Problem in Flexible Manufacturing Systems," Technical Report 89-7, Dept. of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1989b.
- Kimemia J., and S.B. Gershwin, "Flow Optimization in Flexible Manufacturing Systems," *International Journal of Production Research*, vol. 23, no. 1, pp. 81-96, 1985.
- Kiran, A.S., "Complexity of FMS Loading and Scheduling Problems," Working Paper, Dept. of Industrial and Systems Eng., University of Southern California, Los Angeles, 1986.
- Kiran A.S., and B. Tansel, "The System Setup in FMS: Concepts and Formulation," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 321-332, Amsterdam, 1986.
- Kouvelis P., "Design and Planning Problems in Flexible manufacturing Systems," Working Paper Series 1989, Management Dept. Graduate School of Business, The University of Texas at Austin, 1989.
- Kouvelis, P., and H.L. Lee, "Block Angular Structures and the Loading Problem in Flexible Manufacturing System," *Operations Research*, vol. 39, no. 4, pp. 666-676, 1991.
- Kumar R.K., A. Kusiak, and A. Vannelli, "Grouping of Parts and Components in Flexible Manufacturing Systems," *European Journal of Operational Research*, vol 24, pp. 387-397, 1986.

- Kusiak, A., "The Part Families Problem in Flexible Manufacturing Systems," Proceedings of the First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 237-242, Amsterdam, 1984.
- Kusiak, A., "Loading Models in Flexible Manufacturing Systems, Recent Developments in Flexible Manufacturing Systems and Allied Areas," The Design and Operations of FMS, edited by A. Raouf, and S.I. Ahmad, Elsevier, pp. 119-132, New York, 1985a.
- Kusiak, A., "Flexible Manufacturing Systems: A Structural Approach," International Journal of Production Research, vol. 23, no. 1, pp. 81-96, 1985b.
- Kusiak, A., "The Part Families Problem in Flexible Manufacturing Systems," Annals of Operations Research, vol 3, pp. 279-300, 1985c.
- Kusiak, A., "Application of Operational Research Models and Techniques in Flexible Manufacturing Systems," European Journal of Operational Research, vol 24, pp. 336-345, 1986a.
- Kusiak, A., "Scheduling Flexible Machining and Assembly Systems," Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 297-309, Amsterdam, 1986b.
- Kusiak, A., and W.S. Chow, "Efficient Solving of the Group Technology Problem," Journal of Manufacturing Systems, vol.6, no. 2, pp. 117-124, 1987.
- Lashkari, R.S., S.P. Dutta, and A.M. Padhye, "A New Formulation of Operation Allocation Problem in Flexible manufacturing Systems: Mathematical Modelling and Computational Experience," International Journal of Production Research, vol. 25, no. 9, pp. 1267-1283, 1987.
- Maimon, O.Z., and Y.F. Choong, "Dynamic Routing in Reentrant FMS," Robot. Computer Aided Manufacturing, vol. 3, pp. 295-300, 1987.
- Margirier, G., "Flexible Automated Machining in France: Results of a Survey," Journal of Manufacturing Systems, vol. 6, no. 4, pp. 253-265, 1987.
- Mazzola J.B., "Heuristics for the FMS/MRP Rough-Cut Capacity Planning Problem," Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 119-126, Amsterdam, 1989.

- Mazzola J.B., A.W. Neebe, and C.V.R. Dunn, "Production Planning of a Flexible Manufacturing System in a Material Requirements Planning Environment," *The International Journal of Flexible Manufacturing Systems*, vol. 1, pp. 115-142, 1989.
- Moreno, A.A., and F.-Y. Ding, "Goal Oriented Heuristics for the FMS Loading (and Part Type Selection Problems)," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 119-126, Amsterdam, 1989.
- O'Grady, P.J., and U. Menon, "Loading a Flexible Manufacturing System," *International Journal of Production Research*, vol. 25, no. 7, pp. 1053-1068, 1987.
- Ohmi, T., T. Ito, and Y. Yoshida, "Flexible Manufacturing Systems in Japan; Present Status," *Proceedings of the First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 23-29, Amsterdam, 1984.
- Pourbabai, B., "A Production Planning and Scheduling Model for Flexible Manufacturing," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 405-415, Amsterdam, 1986.
- Rachamadugu, R., and K.E. Stecke, "Classification and Review of FMS Scheduling Procedures," *Working Paper, School of Business Administration, The University of Michigan, Ann Arbor*, 1989.
- Rajagopalan, S., "Formulation and Heuristic Solutions for Parts Grouping and Tool Loading in Flexible Manufacturing Systems," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 405-415, Amsterdam, 1986.
- Raman, N., F.B. Talbot, and R.V. Rachamadugu, "Due Date Based Scheduling in a General Flexible Manufacturing System," *Journal of Operations Management*, vol. 8, no. 2, pp. 115-132, 1989.
- Sarin, S.C., and C.S. Chen, "The Machine Loading and Tool Allocation Problem in a Flexible Manufacturing System," *International Journal of Production Research*, vol. 25, no. 7, pp. 1081-1094, 1987.

- Sawik, T., "Modelling and Scheduling of a Flexible Manufacturing System," *European Journal of Operational Research*, vol 45, pp. 177-190, 1990.
- Schriber, T.J., and K.E. Stecke, "Machine utilizations and Production rates Achieved by Using Balanced Aggregate FMS Production Ratios in a Simulated Setting," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 405-415, Amsterdam, 1986.
- Seidmann, A., S. Shalev-Oren, and P.J. Schweitzer, "An Analytical Review of Several Computerized Closed Queueing Network Models of FMS," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 405-415, Amsterdam, 1986.
- Shanker, K., and S. Rajamarthandan, "Loading Problem in FMS: Part Movement Minimization," *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 119-126, Amsterdam, 1989.
- Shanker, K., and A. Srinivasulu, "Some Solution Methodologies for Loading Problems in A Flexible Manufacturing System," *International Journal of Production Research*, vol. 27, no. 6, pp. 1019-1034, 1989.
- Shanker, K., and Y.-J.J. Tzen, "A Loading and Dispatching Problem in a Random Flexible Manufacturing System," *International Journal of Production Research*, vol. 23, no. 3, pp. 579-595, 1985.
- Shantikumar, J.G., and D.D. Yao, "Optimal Server Allocation in a System of Multiserver Stations," *Management Science*, vol. 33, pp. 1173-1180, 1987.
- Sherali, H.D., C. Sarin, and R. Desai, "Models and Algorithms for Job Selection, Routing, and Scheduling in a Flexible Manufacturing System," *Annals of Operations Research*, vol. 26, pp. 433-453, 1990.
- Shirley, G.V., and R. Jaikumar, "Production Planning in Flexible Transfer Lines," *Journal of Manufacturing and Operations Management*, vol. 3, pp. 249-267, 1989.
- Smith, M.L., R. Ramesh, R.A. Dudek, and E.L. Blair, "Characteristics of U.S. Flexible Manufacturing Systems: A Survey," *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and*

Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 321-332, Amsterdam, 1986.

Solberg, J.J., "A Mathematical Model of Computerized Manufacturing Systems," Proceedings of the Fourth International Conference on Production Research, Tokyo, pp. 22-30, 1977.

Solot, P., "A Concept for Planning and Scheduling in an FMS," European Journal of Operational Research, vol 45, pp. 85-95, 1990.

Stecke K.E. "Design, Planning, Scheduling, and Control Problems of Flexible Manufacturing Systems," Annals of Operations Research, vol. 3, pp. 3-12, 1983a.

Stecke K.E. "Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems," Management Science, vol. 29, no. 3, 1983b.

Stecke K.E., "Procedures to Determine Both Appropriate Production Ratios and Minimum Inventory Requirements to Maintain these Ratios in Flexible Manufacturing Systems," Working Paper, no. 448, School of Business Administration, The University of Michigan, Ann Arbor, 1985.

Stecke K.E., "A Hierarchical Approach to Solving Machine Grouping and Loading Problems of Flexible Manufacturing Systems," European Journal of Operational Research, vol 24, pp. 369-378, 1986.

Stecke K.E., "Algorithms to Efficiently Plan and Operate a Particular FMS," Working Paper, no. 566, School of Business Administration, The University of Michigan, Ann Arbor, 1988.

Stecke, K.E., D. Dubois, J. Browne, S.P. Sethi, and K. Rathmill, "Classification of Flexible Manufacturing Systems: Evolution Towards the Automated Factory," FMS Magazine, pp. 114-117, April 1984.

Stecke K.E., and I. Kim, "A Flexible Approach to Implementing the Short-Term FMS Planning Function," Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems: Operations Research Models and Applications, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 321-332, Amsterdam, 1986.

Stecke K.E., and I. Kim, "A Study of FMS Part Type Selection Approaches for Short-Term Production Planning," International Journal of Flexible Manufacturing Systems, vol. 1, pp. 7-29, 1988.

- Stecke K.E., and I. Kim, "Performance Evaluation for Systems of Pooled Machines of Unequal Sizes: Unbalancing versus Balancing," *European Journal of Operational Research*, vol 42, pp. 22-38, 1989.
- Stecke K.E., and T.H. Morin, "The Optimality of Balancing Workloads in Certain Types of Flexible Manufacturing Systems," *European Journal of Operational Research*, vol 20, pp. 68-82, 1985.
- Stecke K.E., and J.J. Solberg, "Loading and Control Policies for a Flexible Manufacturing System," *International Journal of Production Research*, vol. 19, no. 5, pp. 481-490, 1981.
- Stecke K.E., and J.J. Solberg, "The Optimality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multiserver Queues," *Operations Research*, vol. 33, no. 4, pp. 882-910, 1985.
- Stecke K.E., and F.B. Talbot, "Heuristics for Loading Flexible Manufacturing Systems," *Proceedings of the Seventh International Conference on Production Research*, pp. 73-85, Windsor, Ontario, Canada, 1983.
- Stoeva, S.P., "A Due-Date Based Dispatching Rule for Flexible Manufacturing Systems," *International Journal of Production Research*, vol. 28, no. 11, pp. 1991-1999, 1990.
- Suri, R., "An Overview of Evaluative Models for Flexible Manufacturing Systems," *Annals of Operations Research*, vol. 3, pp. 13-22, 1985.
- Suri, R., and R.R. Hildebrandt, "Modelling Flexible Manufacturing Systems Using Mean Value Analysis," *Journal of Manufacturing Systems*, vol. 3, no. 1, pp. 27-39, 1984.
- Suri, R., and C.K. Whitney, "Decision Support Requirements in Flexible Manufacturing," *Journal of Manufacturing Systems*, vol 3, no. 1, pp. 61-69, 1984.
- Talavage, J., and R.G. Hannam, *Flexible Manufacturing Systems in Practice*, Marcel Dekker Inc., New York, 1988.
- Tang C.S., and E.V. Denardo, "Models Arising From a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches," *Operations Research*, vol. 35, no. 5, pp. 767-777, 1988a.
- Tang C.S., and E.V. Denardo, "Models Arising From a Flexible Manufacturing Machine, Part II: Minimization of the Number of Switching Instants," *Operations Research*, vol. 35, no. 5, pp. 778-784, 1988b.

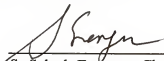
- Venkataramanan, M.A., J. Dinkel, and J. Mote, "A Surrogate and Lagrangian Approach to Constrained Network Problems," *Annals of Operations Research*, vol 20, pp. 283-302, 1989.
- Vinod, B., and J.J. Solberg, "The Optimal Design of Flexible Manufacturing Systems," *International Journal of Production Research*, vol. 23, no. 6, pp. 1141-1151, 1985.
- Whitney, C.K., and T.S. Gaul, "Sequential Decision Procedures for Batching and Balancing in FMSs," *Annals of Operations Research*, vol. 3, pp. 301-316, 1985.
- Whitney, C.K. and R. Suri, "Decision Aids for FMS Part and Machine Selection," *Proceedings of the First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems*, edited by K.E. Stecke and R. Suri, Elsevier Science Publishers B.V., pp. 205-210, Amsterdam, 1984.
- Whitney, C.K., and R. Suri, "Algorithms for FMS Part and Machine Selection in Flexible Manufacturing Systems," *Annals of Operations Research*, vol. 3, pp. 239-261, 1985.
- Wilson, J.M., "An Alternative Formulation of the Operation Allocation Problem in Flexible Manufacturing Systems," *International Journal of Production Research*, vol. 27, no. 8, pp. 1405-1412, 1989.
- Yao, D.D., and J.A. Buzacott, "Modelling the Performance of Flexible Manufacturing Systems," *International Journal of Production Research*, vol. 23, no. 5, pp. 945-959, 1985a.
- Yao, D.D., and J.A. Buzacott, "Modelling a Class of State-Dependent Routing in Flexible Manufacturing Systems," *Annals of Operations Research*, vol. 3, pp. 153-167, 1985b.
- Yao, D.D., and J.A. Buzacott, "The Exponentialization Approach to Flexible Manufacturing System Models with General Processing Times," *European Journal of Operational Research*, vol 24, pp. 410-416, 1986a.
- Yao, D.D., and J.A. Buzacott, "Models of Flexible Manufacturing Systems with Limited Local Buffers," *International Journal of Production Research*, vol. 24, no. 1, pp. 107-118, 1986b.
- Yao, D.D., and J.A. Buzacott, "Modelling a Class of Flexible Manufacturing Systems," with Reversible Routing," *Operations Research*, vol 35, no. 1, pp. 87-93, 1987.

BIOGRAPHICAL SKETCH

Meltem Denizel Sivri received her Bachelor of Science degree in industrial engineering from Middle East Technical University, Turkey, in 1983. In 1986, she received her Master of Science degree in industrial engineering from the same university. She worked as a research assistant in the System Sciences Research Institute, at Middle East Technical University while doing postgraduate work until 1988. She started her Ph.D. studies in decision and information sciences at the University of Florida in 1988. While working on her Ph.D. she also worked as a teaching assistant in the same department.

Meltem Denizel Sivri is a member of ORSA, TIMS, and DSI.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



S. Selcuk Erenguc, Chair
Professor of Decision and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



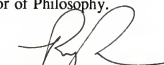
Harold P. Benson
Professor of Decision and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Chung Yee Lee
Associate Professor of Industrial and Systems
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Patrick A. Thompson
Assistant Professor of Decision and Information
Sciences

This dissertation was submitted to the Graduate Faculty of the Department of Decision and Information Sciences in the College of Business Administration and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

May 1993

Dean, Graduate School